STUK-YTO-TR 203 / TOUKOKUU 2004

VAATIMUSHALLINNAN SOVELTAMISMAHDOLLISUUDET YDINTURVALLISUUDEN PARANTAMISESSA SUOMESSA

Markus Renlund, Veli Taskinen

(RAMSE Consulting Oy)

Tutkimuksen yhteyshenkilö Säteilyturvakeskuksessa Tapani Virolainen

STUK • **SÄTEILYTURVAKESKUS**
STRÅLSÄKERHETSCENTRALEN
RADIATION AND NUCLEAR SAFETY AUTHORITY

Osoite/Address • Laippatie 4, 00880 Helsinki
Postiosoite / Postal address • PL / P.O.Box 14, FIN-00881 Helsinki, FINLAND
Puh./Tel. (09) 759 881, +358 9 759 881 • Fax (09) 759 88 500, +358 9 759 88 500 • www.stuk.fi

STUKin raporttisarjoissa esitetyt johtopäätökset ovat tekijöiden johtopäätöksiä, eivätkä ne välttämättä edusta Säteilyturvakeskuksen virallista kantaa.

ISBN 951-712-839-8 (nid.)
ISBN 951-712-840-1 (pdf)
ISSN 0785-9325

Dark Oy, Vantaa 2004

RENLUND Markus, TASKINEN Veli (RAMSE Consulting Oy). Vaatimushallinnan soveltamismahdollisuudet ydinturvallisuuden parantamisessa Suomessa. STUK-YTO-TR 203. Helsinki 2004. 54 s.

Avainsanat: vaatimushallinta, vaatimustenhallinta, ydinturvallisuus, käytäntö, tutkimus, vaatimushallintaprosessi, FIN5, turvallisuusstandardit

Tiivistelmä

Projektin ensimmäisessä vaiheessa kartoitettiin vaatimushallinnan tutkimustilannetta ja soveltuvia käytäntöjä sekä soveltamismahdollisuuksia ydinturvallisuuden parantamisessa Suomessa. Lisäksi kehitettiin vaatimushallintaprosessia Säteilyturvakeskuksen (STUK) uuden ydinvoimalaitoksen (FIN5) valvontatoimintaan.

Vaatimushallintaan liittyvässä tutkimuksessa on kehitetty lähestymistapoja ja menetelmiä, joiden pääpaino on ollut ohjelmistotuotannossa, erityisesti sen vaatimusmäärittelyvaiheessa. Muita vaatimushallinnan osa-alueita, kuten esimerkiksi vaatimusten muutoshallinta, jäljitettävyyden hallinta ja vaatimustietojen hallinta, on tutkittu vähemmän. Ohjelmistojen vaatimushallintamenetelmät ja teknologiat ovat kuitenkin sovellettavissa pitkälti muidenkin järjestelmien vaatimushallintaan. Tietylle sovellusalalle soveltuvia menetelmiä on kehitetty tietoliikennealalle sekä turvallisuuskriittisten ohjelmistojen tuotannolle.

Myös erilaisten käytäntöjen pääpaino on ohjelmistotuotannon vaatimushallinnassa. Selvityksessä on tunnistettu vaatimushallinnan keskeiset lähestymistavat sekä kiinnostavimmat käytännöt lähinnä viranomaisvalvontakäyttöön. Turvallisuuskriittisten järjestelmien kannalta kiinnostaviksi osoittautuivat tavoite- ja näkökulmapohjaiset sekä muodolliset vaatimusmäärittelytavat.

Soveltamismahdollisuuksia viranomaistoiminnassa löytyy uusien ydinvoimalaitosten sekä laitosuusintojen turvallisuusvalvonnassa sekä voimayhtiöissä vastaavasti erilaisissa uus- ja uusintahankinnoissa sekä viranomaislupatoiminnassa.

STUK:n FIN5-valvontatoiminnan tueksi määriteltiin alustava vaatimushallintaprosessi, jota on tarkoitus kehittää STUK:ssa edelleen.

Projektin toisessa vaiheessa kuvattiin menettelytapoja kattavien ja laadukkaiden järjestelmävaatimusmäärittelyiden tuottamiseksi. Lisäksi arvioitiin ydinturvallisuuteen liittyvien standardien sisältämää vaatimushallintaa, mikä osoitti niissä olevan hyvin eri laajuisesti vaatimushallinnan elementtejä. Edelleen kuvattiin tarkemmin vaatimushallinnan tavoite- ja näkökulmapohjaisia lähestymistapoja ja eräillä muilla aloilla sovellettavia vaatimushallintakäytäntöjä.

Yhteenvetona voidaan todeta, että yhtä ainoata, järjestelmien turvallisuuden takaavaa menettelyä vaatimushallintaan ei ole tunnistettu, mutta yhdistämällä vaatimushallinnan erilaisia lähestymistapoja ja käytäntöjä voidaan järjestelmien kehittämisessä päästä korkeaan luotettavuuteen ja turvallisuuteen. Tähän kokonaisuuteen kuuluvat seuraavat keskeisimmät tekijät:

1. Systemaattisen vaatimushallintaprosessin käyttöönotto ja noudattaminen.
2. Tavoite-, näkökulmapohjaisten sekä muodollisten määrittelytapojen käyttö vaatimusten määrittelyyn.
3. Systemaattisten menettelyiden käyttö järjestelmävaatimusmäärittelyiden laadintaan, todentamiseen ja kelpuutukseen.

Sisällysluettelo

1 Johdanto

Vaatimushallinta on systemaattinen lähestymistapa hankkia, organisoida ja dokumentoida järjestelmälle asetetut vaatimukset ja ylläpitää niitä sekä hallita vaatimusten muutoksia järjestelmän koko elinkaaren ajan. Vaatimushallinta on osa projektien tai toiminnan kokonaishallintaa. Sen avulla pyritään useimmiten varmistamaan, että projektissa tuotettava järjestelmä tai kehitettävä toiminta vastaa sidosryhmien tarpeita.

Vaatimushallinnan puutteet on todettu yleisesti tärkeimmäksi projektien epäonnistumisten – projektien aikataulujen venyminen, kustannusten ylittyminen ja tulosten laatupuutteet – syyksi. Systemaattisella vaatimushallinnalla voidaan näitä puutteita vähentää ja poistaa.

Tämän projektin tavoitteena on ollut kartoittaa vaatimushallinnan tutkimustilannetta ja käytäntöjä sekä soveltamismahdollisuuksia erityisesti ydinturvallisuuteen liittyvässä toiminnassa. Samalla on saatu yleistä tietoa vaatimushallinnasta, mitä voidaan hyödyntää muillakin sovellusalueilla.

Projektin ensimmäisessä vaiheessa kartoitettiin vaatimushallinnan tutkimustilannetta ja -käytäntöjä sekä soveltamismahdollisuuksia ja kehitettiin Säteilyturvakeskuksen (STUK) vaatimushallintaprosessia FIN5-ydinvoimalaitoshankkeen valvontatoiminnassa. Ensimmäisen vaiheen tuloksia kuvataan raportissa osissa 2–6.

Projektin toisessa vaiheessa selvitettiin ydinvoiman turvallisuuteen liittyvien (erityisesti automaatioon liittyvät) standardien vaatimushallintaa sekä turvallisuuteen liittyviä vaatimushallintakäytäntöjä eri sovellusalueilla (muut kuin ydinvoima) sekä kuvattiin tarkemmin joitakin projektin ensimmäisessä vaiheessa tunnistettuja lähestymistapoja. Toisen vaiheen tuloksia kuvataan raportissa osissa 7–11. Toiseen vaiheeseen liittyi myös osallistuminen vaatimusmäärittelyiden laadintakurssiin, jonka keskeisin anti kuvataan raportin osassa 7.

2 Vaatimushallinnan tutkimustilanne

2.1 Tutkimustilanneselvityksen tavoitteet

Tämän osaselvityksen tarkoitus on ollut kartoittaa vaatimushallinnan tutkimustilannetta mm. ydinturvallisuuteen liittyvillä sovellusalueilla. Sen piiriin kuuluvat esimerkiksi ydinvoimalaitosten automaation, valvomotoiminnan ja tietotekniikan uudistukset. Tietous vaatimushallinnan tutkimuksesta palvelee vaatimushallinnan kehittämistä ja soveltamista ydinturvallisuuden parantamisessa.

2.2 Tutkimustilanteen kartoitusmenetelmät

Kartoitusta on tehty kahdella hakutavalla: Internet-tietohauilla sekä kyselyllä suomalaisilta vaatimushallinnan asiantuntijoilta. Asiantuntijoiden nimet sekä kyselyn tärkeimmät tulokset on esitetty liitteessä 2-1.

Internet-tietohaulla on haettu konferenssiesitelmiä ja muita Internetissä saatavissa olevia dokumentteja sekä jatkettu hakua edelleen niiden viitedokumenteista. Dokumenttien viiteluettelot ovat yleensä hyvin laajoja, jopa useita kymmeniä viitteitä yhdessä dokumentissa. Viitetiedostoista on tarkastelu kiinnostavimpia, Internetistä tai muuten nopeasti saatavissa olevia dokumentteja.

Tutkimuskartoitusaineistoa on hyödynnetty projektin toisessa osatehtävässä, vaatimushallinnan uusimpien (State–of–the–Art) käytäntöjen tunnistamisessa. Molemmat aihealueet menevät pitkälti limittäin ja selkeää eroa tutkimusten ja käytäntöjen kuvausten välillä ei ole.

2.3 Tutkimustilanneselvityksen rajaukset

Vaatimushallinta on tänään jo niin laaja osaamisalue, että sen käsittely tässä suppeahkossa selvityksessä jää melko yleisluontoiseksi. Tässä projektiosassa oli tavoitteena tunnistaa vaatimushallinnan tutkimustilannetta muutamilla sovellusalueilla, joiksi projektisuunnitelmassa on mainittu automaatiojärjestelmä-, ydinvoimalaitos-, tietoliikenne- ja puolustusalat sekä viranomaistoiminta.

SAFIR-hankkeen alueen tukiryhmässä tunnistettiin kuitenkin tarve fokusoida aluetta ensisijaisesti STUK:n viranomaistoiminnan suuntaan. Selvityksessä tarkastellaankin tästä johtuen lähinnä kansainvälisiä suurimpia tunnistettuja tutkimushankkeita sekä suomalaista tutkimusta STUK:n viranomaistoiminnan tarpeita ajatellen.

2.4 Vaatimushallinnan tutkimus

2.4.1 Suomalainen vaatimushallinnan tutkimus

Suomessa vaatimushallintaan liittyvää tutkimusta on tehty mm. Lappeenrannan teknillisessä yliopistossa (LTY), Joensuun yliopistossa (JOY), Tampereen yliopistossa (TY) ja Teknillisessä korkeakoulussa (TKK).

Erillisistä tutkimushankkeista, joissa on useita osapuolia, merkittävimmät ovat QURE ja Eco-Prop.

QURE-projektin [2-1] tutkimusongelma oli, kuinka organisaatiot voivat kehittää kustannustehokkaasti tuotteita, jotka vastaavat paremmin käyttäjien ja asiakkaiden tarpeita. Tutkimuksessa kehitettiin malleja, menetelmiä ja käytäntöjä vaatimusten määrittelyyn ja hallintaan sekä testattiin näitä malleja ja menetelmiä yhteistyöyritysten kanssa.

Projektissa keskityttiin seuraaviin vaatimusten määrittelyn ja hallinnan osa-alueisiin

- vaatimusten määrittely- ja hallintaprosessien kehittäminen
- vaatimusten hankinta
- vaatimusten priorisointi
- vaatimusten muutosten hallinta
- vaatimusten jäljitettävyys.

Projektissa laadituista raporteista osa on nähtävissä viitteen [2-1] Internet-sivuilla.

Lappeenrannan teknillisessä yliopistossa (LTY) on selvitetty vaatimusmäärittelykäytäntöjä (Requirements Engineering, RE) 12:ssa pienessä ja keskisuuressa ohjelmistoyrityksessä [2-2]. Selvityksen mukaan yritykset eivät ole täysin pystyneet saavuttamaan vaatimusmäärittelyn mahdollistamia hyötyjä. Tärkeimmät kehittämisalueet ovat 1) vaatimusmäärittelyprosessin sovittaminen omaan toimintaan, 2) vaatimusmäärittelyprosessin parantaminen ja 3) vaatimusmäärittelykäytäntöjen automatisointi.

Uolevi Nikula on kehittänyt lisensiaattityössään BARE-menetelmän vaatimusten määrittelyyn ja hallintaan pienissä hallinnollisissa ja liiketoiminnan tietojärjestelmissä [2-3]. Tavoitteena on ollut kehittää yksinkertainen ja helposti käyttöönotettava menetelmä. Se koostuu viidestä perusosasta: vaatimusdokumenttitemplate, vaatimusmäärittely ja -hallintatekniikat, vaatimushallinnan työkalutuki sekä koulutus.

VTT on kehittänyt 1998–99 rakennushankkeiden vaatimushallintaan Excel-pohjaisen työkalun (*EcoProp*), jota on käytetty useissa rakennushankkeissa [2-4]. Sillä voidaan määritellä vaatimukset sekä itse rakennukselle että sen toteutukselle. EcoProP-ohjelma on tarkoitettu toimivuuspohjaisten vaatimusten asettamiseen ennen rakennuksen suunnitteluprosessin aloittamista. Ohjelmaa voi käyttää jo hankesuunnitteluvaiheessa ja hankkeen tavoitteiden lisäys joustavasti myöhemminkin on mahdollista.

VTT Elektroniikka on kehittänyt 1994–96 reaaliaikaisten sulautettujen järjestelmien ohjelmistotuotantoprosessin parantamiseen *Pr²imer*-menetelmän [2-5], joka sisältää myös vaatimushallintatekniikoita.

Suomalaisten tutkimusohjelmien puitteissa on tehty useampia väitöskirjoja ja tuotettu useita julkaisuja vaatimushallinnan eri osa-alueista. Liitteessä 2–3 on lueteltu joitakin automaatiojärjestelmien vaatimushallintaan liittyviä suomalaisia dokumentteja ja IEC-standardeja.

2.4.2 Kansainvälinen vaatimushallinnan tutkimus

Lähteessä [2-6] on ansiokas katsaus *vaatimusmäärittelyn tutkimuksesta* viimeisten 25 vuoden ajalta. Siinä on kuvattu myös vaatimushallinnan pääkäsitteet ja tekniikat, jotka on kehitetty tähän päivään mennessä vaatimusmäärittelyn tehtävien tueksi. Monimutkaista turvallisuuskriittistä järjestelmää on käytetty esimerkkinä esiteltäessä monia tutkimustrendejä vaatimusmäärittelyn alueella.

Dokumentissa korostetaan vaatimusmäärittelyn keskeistä merkitystä järjestelmien kehittämisessä. Dokumentin kirjoittajan suosikki monista vaatimusmäärittelylähestymistavoista on tavoitepohjainen vaatimusmäärittely. Se soveltuu käytettäväksi vaatimusmäärittelyn kaikissa vaiheissa – mm. vaatimusten selvittämisessä, ristiriitojen selvittämisessä, poikkeustilanteiden tunnistamisessa ja myös arkkitehtuurin johtamisessa vaatimuksista.

Viitteessä [2-7] on kuvattu mm. *vaatimusmäärittelyn ongelmia ja vaatimusten selvityksen ja neuvottelun* (negotiation) tutkimusta. Vaatimusten selvityksen ongelmat jaetaan siinä järjestelmän rajojen ongelmiin, sidosryhmien ja analysoijan väliseen ymmärtämiseen liittyviin ongelmiin sekä vaatimusten muuttumiseen liittyviin ongelmiin. Viitteessä on esitetty lukuisia näiden ongelmien käsittelyyn liittyviä tutkimuksia. Samoin on käsitelty ristiriitaisten vaatimusten ratkaisuun liittyviä vaatimusten neuvottelun tutkimuksia.

REAIMS-projektin [2-8] (ESPRIT, 1994–96) tavoitteena oli kehittää viitekehys vaatimusmäärittelyprosessin parantamiseksi luotettavien ja turvallisuuskriittisten järjestelmien kehittämisessä sekä arvioida sitä todellisessa teollisuusympäristössä. Projektissa luotiin mm. vaatimusmäärittelyprosessin kypsyysmalli nykyisten käytäntöjen arviointiin, vaatimusmäärittelyn hyvien käytäntöjen ohje sekä prosessin parantamismalli. Siinä kehitettiin myös Näkökulma-pohjainen (Viewpoint) vaatimusmäärittelymalli. Hyvien käytäntöjen ohjeessa on yli 50 suositeltua käytäntöä vaatimushallinnan eri vaiheisiin. Projektin tuloksia on kuvattu myös lähteessä [2-9].

Luodut käytännöt ja prosessin parantamismalli sopivat erityisesti monimutkaisten ohjelmistojärjestelmien suunnitteluun, joilta vaaditaan suurta luotettavuutta ja jotka ovat usein turvallisuuskriittisiä. Sovellusalueita ovat mm. pankki- ja vakuutusala sekä telekommunikaatio-, kuljetus-, avaruus- ja kemiallinen teollisuus. Käytännöt soveltuvat myös muiden alojen vaatimushallintaan.

KAOS-menetelmä [2-10] sisältää sekä vaati-

musten kuvauskielen että tavoitepohjaisen (goal-driven elaboration) menetelmän vaatimusten työstämiseen samoin kuin agenttien käytön vaatimusmäärittelyssä. Se sisältää myös metatason tietämystä vaatimusten löytämisen helpottamiseksi. Menetelmä sisältää myös Miksi-aspektin perinteisen Mitä-aspektin lisäksi. Mitä-aspekti on perinteisen vaatimusmäärittelyn sidosryhmävaatimusten näkökulma; sidosryhmävaatimuksilla ilmaistaan, *mitä* käyttäjän on järjestelmällä voitava tehdä tai aikaansaada. Miksi-näkökulmalla saadaan perustelut vaatimuksille ja kehitettävälle järjestelmälle.

KAOS-menetelmän avulla voidaan aikaansaada kattavia ja oikeiksi todistettuja vaatimusmäärittelyjä. Menetelmä sisältää mm. työkalut tavoitteiden jakamiseksi osatavoitteiksi sekä siitä edelleen vaatimuksiksi. KAOS-mallin ja sen kuvauskielen avulla voidaan kuvata vaatimuksia *tavoitteiden, rajoitusten, objektien, toimenpiteiden, agenttien, suhteiden, tapahtumien yms.* avulla. Kuvauskieli on kaksitasoinen: toisella tasolla kuvataan käsitteitä, niiden attribuutteja sekä liityntöjä, toisella määritellään käsitteet muodollisella kuvaustavalla. Vaatimusten välisten liitäntöjen kuvaamiseen on myös esitystavat.

KAOS-menetelmä on tunnetuimpia tavoitepohjaisen vaatimusmäärittelyn menetelmiä. Se on muodostanut perustan monille jatkosovelluksille. Viitteessä [2-11] on myös kuvattu tavoitepohjaista vaatimusmäärittelyä.

ICARUS-tutkimuksessa (ESPRIT (1989–94) [2-12] kehitettiin ns. agenttipohjaista vaatimusten esittämistapaa ja skenaario-animaatiota ohjelmistoprojekteissa. Projektissa kehitettiin muodollinen vaatimusten kuvauskieli, jolla vaatimukset voidaan kuvata ja järjestää. Myös vaatimusten tuottamisprosessia kehitettiin.

NATURE-tutkimuksessa [2-13] (1992–95) luotiin kattava vaatimushallintaviitekehys ja -teoria sovellusala-analyysiä, prosessin mallintamista sekä vaatimustiedon hallintaa varten.

Taustana tutkimukselle oli vaatimusmäärittelyn muuttuminen projektien alkuvaiheen tukemisen sijasta kattamaan monimutkaisten, pitkäikäisten, nopeasti muuttuvassa toimintaympäristössä toimivien ihminen–kone-järjestelmien koko elinkaari. NATURE-viitekehyksen mukaan vaatimusmäärittely on jatkuva prosessi, joka kehittää eri sidosryhmien visiot ja vaatimukset monimutkaisissa yhteyksissä ja ympäristöissä.

NATUREssa kehitettiin kolme erityisteoriaa. Vaatimussovellusalue-teoria (domain theory) antaa opastusta siitä, mikä asiayhteystietämys on tarpeen ja kuinka se on organisoitava. Vaatimusprosessiteoria tarjoaa yhtenäisen prosessi-metamallin, jossa pieni joukko rakenne-elementtejä kattaa laajemman ja joustavamman valikoiman prosessin ohjausstrategioita kuin muut ohjelmisto- tai työnkulkumallit. Tietämyksen esitysteorian tarkoituksena on määritellä, mitä alakohtaista ja prosessitietämystä on hankittava ja kuinka sitä on hallittava käyttäen epämuodollisten, puolimuodollisten ja muodollisten vaatimusten esitystapojen yhdistelmää.

Projektissa kehitettiin myös prototyyppityökaluja kehitettyjen teorioiden testaamista varten. Projektin raportit ovat saatavissa Internetissä [2-13].

CREWS-projekti [2-14] oli ESPRIT-ohjelmaan (1996–99) kuuluva tutkimus, jossa kehitettiin ja arvioitiin menetelmiä ja työkaluja skenaariopohjaiselle vaatimusten tunnistamiselle ja kelpuutukselle. Skenaariolähestymistapa on hyvin yleinen vaatimusmäärittelytapa, jota käytetään monien menetelmien osana. Projektin raportit sekä templatet ovat saatavissa Internetistä.

ARENA-projektissa [2-15] kehitettiin vaatimusten analysointimenetelmä tietoliikennepalvelujärjestelmien suunnitteluun. ARENA on vaatimusmäärittelyprosessi, joka jakaa vaatimukset yleisiin, erityisiin ja muodollisiin vaatimuksiin. Se sisältää myös oliopohjaisen tietoliikennejärjestelmän palvelumallin. Sen tärkein tavoite on varmistaa vaatimusten jäljitettävyys. Menetelmässä käytetään myös QFD-menetelmää (Quality Function Deployment) asiakasvaatimusten tunnistamiseen.

RATS-projektissa [2-16] (Requirements Acquisition and Specification for Telecommunication Services, 1998) kehitettiin menetelmä ja työkalu korkealaatuisten tietoliikennepalvelujen suunnitteluun. Lopputuloksena on RATS-asiantuntijajärjestelmään pohjautuva työkalu, joka ohjaa käyttäjää koko järjestelmän kehitysprojektin ajan. RATS-työkalulla voi toteuttaa vaatimushallinnan, vaatimusten jäljitettävyyden hallinnan ja vaikutusanalyysin sekä vaatimusten dokumentoinnin.

IMPRESSION-projektin [2-17] tavoitteena oli osoittaa vaatimushallinnan parhaiden käytäntöjen hyödyt ohjelmistokehityksessä. Projektin tulokset osoittavat, että strukturoidun vaatimusmäärittelylähestymistavan soveltaminen, sisältäen käytäntöjen käyttöönoton ja koulutuksen, johtaa vaatimusmäärittelyn kyvykkyyden selvään parantumiseen.

Viitteessä [2-18] on kuvattu *vaatimusten vuorovaikutusten hallintaan* (Requirements Interaction Management) liittyviä tutkimuksia. Vuorovaikutusten hallinta tarkoittaa niitä toimenpiteitä, joilla tunnistetaan, hallitaan ja järjestetään järjestelmän vaatimusten ja vaatimusjoukkojen kriittisiä suhteita. Tutkimuksissa vuorovaikutusten hallinta on todettu vaatimushallinnan kriittiseksi osa-alueeksi, joka osaltaan johtaa vähempiin virheisiin ja korkeampaan sidosryhmätyytyväisyyteen. Vuorovaikutusten tutkimus on lisääntymään päin.

Vaatimusten muutokset järjestelmäprojektin aikana ovat suuri ongelma mm. tietokonepohjaisten järjestelmien suunnittelussa ja toteutuksessa. Nykyiset vaatimusmäärittelymenetelmät nojaavat prosessipohjaisiin menetelmiin, mistä puuttuvat tuotenäkökulmaan liittyvät ominaisuudet. Viitteessä [2-19] on kuvattu selvitystä, jossa on tutkittu kahden casen kautta vaatimusten muutoksia. Selvitys painottaa vaatimusmäärittelyn lähestymistavan muutosta *tuotepohjaiseksi*, jolla vaatimusten muutoksia voidaan hallita paremmin esimerkiksi turvallisuuskriittisten sovellusten suunnittelussa.

Vaatimusmäärittelyn mallinnusta on kuvattu viitteessä [2-20]. Dokumentissa esitellään vaatimusmäärittelyn uudenlaisia konseptimalleja. Viitteessä [2-21] on annettu yleiskuva vaatimusmäärittelystä ml. mallinnus.

2.5 Menossa olevia vaatimushallinnan tutkimuksia

Viitteessä [2-22] mainitussa lähteessä on kuvattu Martin Glinzin johdolla menossa olevia vaatimushallinnan tutkimuksia, jotka koskevat pääasiassa *vaatimusten määrittelykieliä ja skenaarioiden* käyttöä.

Parhaillaan menossa oleva *CORE-hanke* [2-23] on vuoden 2003 alussa käynnistetty tutkimusprojekti. Sen rahoittavat TEKES ja 10 yhteistyöyritystä. Projektin tavoitteena on kehittää systemaattisia käytäntöjä, joiden avulla suomalaiset organisaatiot voivat kustannustehokkaasti ottaa sidosryhmiä mukaan tuotekehitykseen niin, että tuotteet täyttävät asiakkaiden ja käyttäjien tarpeet. Tämän tavoitteen saavuttamiseksi projektissa kehitetään ja sovelletaan prosesseja, käytäntöjä ja työkaluja sekä tuetaan niiden käyttöönottoa yhteistyöyrityksissä. Projekti jakautuu kolmeen osa-alueeseen:

- Kuinka yhdistää sidosryhmien tietämystä innovaatioiden lisäämiseksi?
- Kuinka analysoida heterogeenisia asiakas- ja käyttäjätarpeita kilpailukykyisten ominaisuuksien valitsemiseksi?
- Kuinka identifioida ja kuvata tuoteperheitä?

Viitteessä [2-24] on kuvattu *LTY:ssä* menossa olevaa Uolevi Nikulan väitöskirjatyötä, jossa kehitetään pienille kaupallishallinnollisille ohjelmistoprojekteille sopivaa vaatimusmäärittelykäytäntöä, joka pohjautuu Jacksonin Simple IS-menetelmään. Tavoitteena on löytää kevyempi vaihtoehto vaatimushallintakäytännöille, jotka yleensä on kehitetty suurten ja monimutkaisten projektien vaatimusmäärittelyyn.

REDEST-projektissa [2-25] (2001–) tavoitteena on kehittää 14 eurooppalaisen ohjelmistoyritysten kanssa sulautettujen järjestelmien innovatiivisia vaatimusmäärittelymenetelmiä ja työkaluja sekä ottaa ne käyttöön projekteissa. Se keskittyy pääasiassa vaatimusten keräysvaiheen menetelmäkehitykseen.

Projektin tarkoituksena on parantaa eurooppalaisen tieteen ja teknologian laatua. Lisäksi REDEST:ssä mukana olevien yritysten muodostamasta ryhmästä on aiottu muodostaa riippumaton ja tunnustettu vaatimusmäärittelyyn liittyvän tiedon lähde.

2.6 Tunnistettuja vaatimushallinnan tutkimustarpeita

Viitteessä [2-26] tarkastellaan vaatimushallintaa käsittäen vaatimusten ja niihin liittyvien tietojen järjestämisen sekä vaatimusten muutosten hallinnan. Siinä todetaan, että näillä tärkeillä alueilla on hyvin vähän tutkimusta. Kirjoittaja esittääkin viisi tutkimusaluetta, missä lisätutkimus on tarpeen: vaatimusten muutosten hallinta, vaati-

musten keskinäiset riippuvuudet, vaatimushallintatyökalut, vaatimuksiin liittyvät tiedot sekä piilevät, ei-ilmaistut vaatimukset.

2.7 Vaatimushallinnan tutkimustilanne, yhteenveto

Vaatimushallinnan suurissa tutkimushankkeissa on kehitetty käyttökelpoisia peruslähestymistapoja ja menetelmiä, joita on myöhemmin kehitetty edelleen ja sovellettu eri alojen projekteissa. Pääpaino tutkimuksissa on ollut ja on edelleen ohjelmistotuotannossa ja siinä nimenomaan ohjelmistojen vaatimusmäärittelyvaiheessa. Vaatimusmäärittelyn merkitys järjestelmäsuunnittelussa sekä sen vaikeus on ohjannut tutkimusta kehittämään menetelmiä nimenomaan sitä vaihetta varten. Muita vaatimushallinnan osa-alueita – kuten esimerkiksi vaatimusten muutoshallinta, jäljitettävyyden hallinta, vaatimustietojen hallinta – on tutkittu vielä varsin vähän.

Muiden järjestelmien kuin ohjelmistojen vaatimushallintaa on tutkittu selvästi vähemmän. Ohjelmistojen vaatimushallintamenetelmät ja teknologiat ovat kuitenkin sovellettavissa hyvin pitkälti muidenkin järjestelmien vaatimushallintaan.

Menossa on hyvin suuri määrä erilaisia vaatimushallinnan osa-alueiden tutkimuksia. Ne kohdistuvat hyvin moniin eri osa-alueisiin pääpainon ollessa edelleenkin ohjelmistotuotannossa.

Selvityksen osatavoitteena oli tunnistaa erityisiä, suoraan sovellusalakohtaisiin vaatimushallintakäytäntöihin liittyviä tutkimuksia. ARENA- ja RATS-tutkimukset oli kohdennettu tietoliikennealalle sekä REAIMS turvallisuuskriittisten ohjelmistojen kehittämiseen. Muissa tutkimuksissa ei erityistä sovellusalaa pidetty lähtökohtana.

Tutkimuksiin liittyneiden case-tapauksia analysoimalla on mahdollista löytää lisää erityisiä, tietyille sovellusaloille sopivia käytäntöjä. Yleisesti voidaan todeta, että kehitetyt käytännöt soveltuvat monille sovellusalueille.

Osa 2 – Viiteluettelo

2-1 QURE-tutkimusprojekti (Quality Through Requirements, 1999–2002, TEKES-projekti), http://www.soberit.hut.fi/qure/suomi/

2-2 Uolevi Nikula, Jorma Sajaniemi, Heikki Kälviäinen, A State–of–the–Practice Survey on Requirements Engineering on Small- and Medium-Sized Enterprises, 2000, ISBN 951-764-431-0, http://www.cs.ucl.ac.uk/research/renoir/links.html

2-3 Nikula, U., BaRE – A Ready to Use Method for Requirements Engineering, in Department of Information Technology. Licentiate Thesis, 2002, Lappeenranta University of Technology: Lappeenranta. pp. 77, http://www.lut.fi/~unikula/

2-4 EcoProp, käyttöohje, http://cic.vtt.fi/eco/ecoprop/

2-5 Pr²imer. http://mango2.vtt.fi:84/ele/research/soh/products/primer/primer.htm

2-6 Axel van Lamsweerde, Requirements engineering in the year 00: a research perspective, Proceedings of the 22nd international conference on Software engineering, p. 5–19, June 04–11, 2000, Limerick, Ireland, http://citeseer.nj.nec.com/vanlamsweerde00requirements.html

2-7 Daniela E. Herlea Damian, Challenges in Requirements Engineering, http://pharos.cpsc.ucalgary.ca/Dienst/Ul/2.0/Describe/ncstrl.ucalgary_cs/1999-645-08

2-8 Pete Sawyer, Ian Sommerville and Stephen Viller, Improving the Requirements Process, http://citeseer.nj.nec.com/471055.html

2-9 Sommerville, I., Sawyer, P.: Requirements Engineering – A Good Practice Guide, John Wiley, 1997, ISBN 0 471 97444 7

2-10 Goal-Driven Requirements Engineering: the KAOS Approach, http://www.info.ucl.ac.be/research/projects/AVL/ReqEng.html

2-11 Axel van Lamsweerde, Building Formal Requirements Models for Reliable Software, ftp://info.ucl.ac.be/pub/publi/2001/avl-AdaEurope.pdf

2-12 Incremental Construction and Reuse of Requiremens Specifications, ICARUS-2537, http://www.newcastle.research.ec.org/esp-syn/text/2537.html

2-13 Novel Approaches to Theories Underlying Requirements Engineering, ESPRIT, 1992–95, http://www-i5.informatik.rwth-aachen.de/PROJEKTE/NATURE/nature-reps-english.html

2-14 CREWS, http://sunsite.informatik.rwth-aachen.de/CREWS/reports.htm

2-15 ARENA Requirements Engineering in a Telecommunication Environment. Uppsala University, August 1997, http://www.docs.uu.se/docs/fi/arena/main.html

2-16 Armin Paul-Gerhard Eberlein, Requirements Acquisition and Specification for Telecommunication Services, http://www.aaai.org/Magazine/Dissertations/1998/eberlein.html

2-17 Impression, the results, http://www.atc.gr/impression

2-18 William N. Robinson, Suzanne D. Pawlowski, Vecheslav Volkov, Requirements Interaction Management, http://citeseer.nj.nec.com/robinson99requirement.html

2-19 Stuart Anderson and Massimo Felici, Requirements Evolution From Process to Product Oriented Management, http://www.dirc.org.uk/publications/papers/9.pdf.

2-20 Colette Rolland, Naveen Prakash, From Conceptual Modelling to Requirements Engineering, http://crinfo.univ-paris1.fr/DEA_I3/papier.pdf

2-21 Klaus Pohl, Requirements Engineering: An Overview, ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-96-02.pdf

2-22 Overview, Research Activities & Projects, Requirements Engineering Research Group (prof. Martin Glinz), http://www.ifi.unizh.ch/groups/req/projects/rerg_projects.html

2-23 CORE (Competitive Advantage through Stakeholder-Driven Requirements Engineering), 2003–, TEKES project, http://www.soberit.hut.fi/core/

2-24 Uolevi Nikula, Minimum Requirements Engineering: An Approach To Technology Transfer For Small Projects, http://www.it.lut.fi/opetus/00-01/010976000/seminars/Nikula.pdf

2-25 REDEST, http://www.redest.net

2-26 Åsa Grehag, Requirements Management in a Life Cycle Perspective – A Position Paper, http://www.ifi.uib.no/conf/refsq2001/papers/p26.pdf

Osa 2, Liite 1 Tutkimustilannekyselyn asiantuntijat ja vastausten tiivistelmät

Uolevi Nikula, LTY

- Menossa on väitöskirjatutkimus vaatimusmäärittelystä: käyttövalmiin (sovellusaluekohtaisen) VM-menetelmän toimivuudesta käytännössä. Sovellusalueena pienissä projekteissa toteutettavat kaupallishallinnolliset sovellukset. Ohjaajina ovat prof. Jorma Sajaniemi, JOY ja prof. Heikki Kälviäinen, LTY.

Osmo Vikman, Nokia Research Centre

- vaatimushallinnan internet-linkkejä ja konferenssiviitteitä (IEEE ym.)
- viittaukset QURE:een, CORE:een ja LTY:n tutkimuksiin.

Pasi Pasivirta, Puolustusvoimat

- Puolustusvoimissa on kehitetty oma vaatimushallintaprosessi hankeohjausprosessin yhteyteen.
- Puolustusalan vaatimushallintaa tutkittu ja kehitetty esim. UK:ssa.
- Tiedot eivät yleensä ole julkisia.

Jorma Sajaniemi, JOY
- tutkimusta tehdään LTY:n kanssa yhteistyössä (U Nikula).

Marjo Kauppinen, SoberIT
- QURE projektissa lähtökohtana oli ohjelmistointensiiviset tuotteet ja siinä kehitettiin monia vaatimushallinnan käytäntöjä. Osa niistä pohjautuu kirjaan Sommerville, Sawyer, Requirements Engineering – A Good Practice Guide.
- Menossa CORE-tutkimusprojekti (2003–2005), jonka tarkoituksena on kehittää käytäntöjä, joiden avulla suomalaiset organisaatiot voivat kustannustehokkaasti ottaa sidosryhmiä mukaan tuotekehitykseen siten, että tuotteet täyttävät asiakkaiden ja käyttäjien tarpeet.

Erkki Knuuttila, Merivoimat
- viittasi vastauksessaan em. suomalaisiin tutkimuksiin sekä RAMSEn asiantuntijoihin.

Timo Käkölä, JYY
- muutamia tiedostoviitteitä.

Osa 2, Liite 2 Vaatimushallinnan termistöä

Tässä selvityksessä käytetyt yleisimmät suomalaiset vaatimushallintaprosessiin liittyvät termit sekä niiden englanninkieliset vastineet:

Vaatimus(ten)hallinta
Requirements Management

Vaatimus(ten)määrittely
Requirements Engineering

Vaatimusten selvitys/ löytäminen
Requirements Elicitation

Vaatimusten kehittyminen
Requirements Evolution

Vaatimusten todentaminen
Requirements Verification

Vaatimusten kelpuutus
Requirements Validation

Vaatimusten muutoshallinta
Requirements Change Management

Vaatimusten jäljitettävyys
Requirements Traceability

Vaatimustietojen hallinta
Requirements Information Management

Osa 2, Liite 3 Automaation vaatimusmäärittelyyn ja kelpoistukseen liittyvää kirjallisuutta

Laatu automaatiossa. Parhaat käytännöt. Suomen Automaatioseura, 2001.

Tommila, Viitamäki: Vaatimusmäärittely prosessiautomaatiossa. Lähestymistapoja esi- ja perussuunnitteluun. VTT tiedotteita 1292. Espoo 1991.

Kallela, Tommila, Tuominen: Osallistuva automaation kehittäminen. Menetelmät. VTT tiedotteita 1511. Espoo 1993.

Tommila, Toola, Viitamäki: Prosessin mallintaminen ohjausjärjestelmän suunnittelun lähtökohtana. VTT tiedotteita 1099. Espoo 1990.

PSK 4601. Automaatiojärjestelmän hankinta. Hankinta-asiakirjat. Prosessiteollisuuden standardoimiskeskus, 1986.

IEC 61508. Functional safety of electrical/electronic/programmable electronic safety-related systems.

IEC 60880. Software for computers in the safety systems of nuclear power stations.

IEC 61513. Nuclear power plants – Instrumentation and control systems important for safety – General requirements for systems

3 Vaatimushallintakäytännöt

3.1 Johdanto

Järjestelmien parannukset ja muutokset sekä uusien järjestelmien hankinnat toteutetaan projekteina. Vaatimushallinta on osa projektien hallintaa ja se vastaa projekteissa siitä, että lopputulos vastaa haluttua eli että laatu (ml. turvallisuus) on halutunlaista.

Vaatimushallinnan käytäntöjä voidaan käyttää myös jatkuvaluonteisen toiminnan, kuten esimerkiksi koko yrityksen tai sen eri toimintojen ohjaukseen. Tällöin voidaan puhua vaatimusohjatusta toiminnasta.

Vaatimushallinta on tänä päivänä jo erittäin laaja osaamisalue, jolle on kehitetty lukuisa määrä erilaisia toteutustapoja eli käytäntöjä, kuten esimerkiksi mallinnus- ja lähestymistapoja, prosesseja, kuvaustapoja yms. Vaatimushallinnan soveltajan onkin osattava valita omaan tarkoitukseensa parhaiten sopivat käytännöt.

Tämän osavaiheen tarkoituksena on ollut kartoittaa uusia (State–of–the–Art) vaatimushallintakäytäntöjä, jotka sopivat käytettäväksi ensisijaisesti ydinturvallisuuden hallintaan liittyvissä hankkeissa, projekteissa ja muussa toiminnassa.

3.2 Vaatimushallinnan käytäntöjen selvityksen rajaukset

Vaatimushallinta on laaja osaamisalue, jonka kokonaisvaltainen tarkastelu vaatii huomattavan ajan ja työmäärän. Tässä projektissa on ollut tarkoitus tunnistaa State–of–the–Art-vaatimushallintakäytäntöjä muutamalla sovellusalueella, joiksi projektisuunnitelmassa on mainittu automaatiojärjestelmä-, ydinvoimalaitos-, tietoliikenne- ja puolustusala sekä viranomaistoiminta. SAFIR-hankkeen alueen 4 tukiryhmän kokouksessa 4.4.03 tunnistettiin tarve fokusoida aluetta. Sen mukaan STUK:n tarpeet ydinturvallisuuden varmistajana ovat ensisijalla.

3.3 Vaatimustenhallintaan liittyviä käsitteitä

Vaatimus(ten)hallinta, VH (Requirements Management) käsittelee hankittavan tai kehitettävän järjestelmän tavoitteiden määrittelyä, näiden tavoitteiden kääntämistä vaatimusmäärittelyiksi sekä vaatimusten kohdentamista järjestelmän osille [3-1].

Sillä tarkoitetaan yleisimmin kaikkea sitä toimintaa, jolla määritellään, organisoidaan ja dokumentoidaan vaatimukset järjestelmän (joka voi sisältää laitteita, ohjelmistoa ja ihmisiä) aikaansaamiseksi, todennetaan (verifioidaan) nämä vaatimukset ja kelpuutetaan (validoidaan) valmis järjestelmä vaatimuksia vastaan sekä hallitaan kaikkea vaatimuksiin liittyvää tietoa ja niiden muutoksia. Vaatimushallinnan tärkeimmät osaalueet ovat vaatimusmäärittely ja vaatimustietojen sekä vaatimusten muutosten hallinta.

Vaatimusmäärittely, VM (Requirements Engineering, RE) on osa vaatimushallintaa, mikä kattaa vaatimusten aikaansaamiseen liittyvän osuuden.

Lähteessä [3-2] on esitetty useita vaatimusmäärittelyn määritelmiä, joista seuraavana muutamia:

1) Vaatimusmäärittelyiden (spesifikaatioiden) tuottamisprosessi.
2) Järjestelmällinen ja iteratiivinen, yhteistyössä tehtävä vaatimusten kehittämisprosessi ongelman analysoimiseksi, tuloshavaintojen dokumentoimiseksi eri esitysmuodoissa sekä tulkinnan virheettömyyden tarkistamiseksi.
3) Järjestelmän, laitteen tai ohjelmiston vaatimusten selvittämis- ja jalostamisprosessi (IEEE-610.12, 1991).

Lähteessä [3-3] on puolestaan seuraava määritelmä: Vaatimusmäärittely on ohjelmistosuunnitte-

lun (järjestelmäsuunnittelun) haara, joka käsittelee ohjelmistojen todellisia tavoitteita, toimintoja ja reunaehtoja. Se käsittelee myös näiden tekijöiden suhteita tarkoituksena määritellä tarkasti ohjelmiston käyttäytyminen sekä sen ajallinen ja ohjelmaperheen kehittyminen.

Perinteisesti vaatimusmäärittelyllä on haettu vastauksia kysymykseen mitä järjestelmältä odotetaan. Nykyaikainen vaatimusmäärittely käsittelee kolmenlaisia kysymyksiä: Miksi-Mitä-Kuka-kysymyksiä [3-1]. Miksi-kysymyksillä haetaan perustelut vaatimuksille, Mitä-kysymyksillä itse vaatimukset ja Kuka/mikä-kysymyksillä järjestelmän toimijat – ns. agentit – sekä vaatimusten liitynnät niihin.

Vaatimusten muutosten hallinta, (Requirements Change Management) käsittää vaatimusten muutoksiin liittyvät toiminnat ml. vaatimusten jäljitettävyyden hallinnan. Vaatimukset eivät ole pysyviä vaan ne voivat muuttua projektin aikana useastakin syystä. Muutosten vaikutukset projektiin on pystyttävä selvittämään nopeasti ja kattavasti, ja tässä auttaa vaatimusten jäljitettävyys. Muutosten teon on oltava hallittua, jotta kuka tahansa ei voisi ilman perusteita tehdä muutoksia, jotka voivat vaikuttaa merkittävästi projektin onnistumiseen.

Vaatimusten jäljitettävyydellä, (Requirements Traceability) tarkoitetaan yleensä kykyä kuvata ja seurata vaatimuksen elinkaarta sekä eteen- että taaksepäin (esim. vaatimuksen lähteestä läpi järjestelmän kehitys-, suunnittelu-, toteutus- ja käyttövaiheiden sekä vaatimuksen jalostamis- ja iteraatiovaiheiden) [3-4]. Lähteessä [3-4] on tarkasteltu jäljitettävyyteen liittyviä ongelmia ja esitetty parannuksia jäljitettävyyden hallintaan.

3.4 Vaatimushallintakäytännöistä

Vaatimushallintakäytännöiksi katsotaan tässä selvityksessä erilaiset vaatimushallinnan prosessit, lähestymistavat ja viitekehykset, vaatimushallintaprosesseissa käytetyt menetelmät ja mallinnustavat, vaatimusten kuvaustavat ja kuvauskielet sekä vaatimusten katselmointi-, todentamis- ja kelpuutustavat. Selkeää eroa esim. prosessien, lähestymistapojen ja menetelmien välillä ei kuitenkaan aina ole nähtävissä, sillä esitetyt kuvaukset sisältävät piirteitä useista eri näkökulmista.

3.5 Vaatimusmallinnus

Mallinnus on vaatimushallinnan ydinprosessi [3-5]. Olemassa oleva ja suunniteltava järjestelmä mallinnetaan ensin jollain abstraktilla tavalla, minkä perusteella järjestelmä suunnitellaan ja toteutetaan. Vaatimusmallinnus palvelee vaatimusmäärittelyä samoin kuin vaatimusten dokumentointia ja muutoksia. Vaatimusmäärittelyn tutkimus ja menetelmäkehitys onkin keskittynyt pitkälti vaatimusten mallinnus- ja määrittelytekniikoihin.

Mallinnus tarkoittaa abstraktia kuvausta suunniteltavasta järjestelmästä [3-6]. Malleilla voidaan kuvata monia vaatimusmäärittelyprosessin osa-alueita, kuten esimerkiksi vaatimusten selvittämistä. Vaatimusmäärittelyyn liittyviä tärkeitä mallinnusalueita ovat viitteen [2-6] mukaan yritysmallinnus (enterprise modeling), datamallinnus, käyttäytymisen (behavior) mallinnus, sovellusalamallinnus (domain modeling) sekä ei-toiminnallisten vaatimusten mallinnus.

Yritysmallinnuksella kuvataan organisaation käyttäytymistä ja samalla saadaan tietoa kehitettävän järjestelmän tarkoituksesta sekä liitynnöistä yrityksen toimintaan. Mallinnustapoja on käytössä monenlaisia, joista erästä tietojärjestelmämallinnukseen liittyvää lähestymistapaa on kuvattu lähteessä [3-7]. Tuloksena saadaan mm. liiketoimintatavoitteet, joita voidaan käyttää suoraan vaatimusmäärittelyn lähtötietoina.

Datamallinnusta käytetään yleisesti suurten tietojärjestelmien mallinnuksessa. Mallinnusmenetelmistä ovat esimerkkeinä kohdemallit (entity-relationship) sekä nykyään yhä yleisemmin käytetyt monet oliopohjaiset (object-oriented) menetelmät.

Käyttäytymisen mallinnuksella kuvataan järjestelmän dynaamista ja toiminnallista käyttäytymistä. Laaja joukko erilaisia mallinnusmenetelmiä alkaen strukturoidusta analyysistä oliopohjaisiin menetelmiin sekä epämuodollisista muodollisiin kuvaustapoihin on käytössä.

Sovellusalamallinnuksella kuvataan järjestelmän toimintaympäristöä.

Ei-toiminnallisten vaatimusten (laatuvaatimusten) mallinnuksella vaatimukset pyritään kuvaamaan mitattavassa tai testattavassa muodossa.

Vaatimusmäärittelyprosessin mallintamista on kuvattu esimerkiksi viitteessä [3-8].

3.6 Vaatimushallintaprosessi

Prosessi on organisoitu joukko toimia, jotka muuttavat lähtötiedot lopputuloksiksi [3-9]. Vaatimushallinnassa on esittäjästä ja painotuksesta riippuen käytössä erilaisia prosesseja, joista seuraavassa on esitetty esimerkkejä.

3.6.1 Erilaisia vaatimushallintaprosesseja

Vaatimushallintaprosessiin kuuluvat viitteen [2-9] mukaan seuraavat osaprosessit:

Soveltuvuusselvitys, jolla selvitetään, kannattaako ajateltu idea (uusi järjestelmä tms.) toteuttaa.

Vaatimusten selvitys ja analyysi, jossa tuotetaan hankittavalle tai suunniteltavalle järjestelmälle, tuotteelle tai organisaatiolle asetettavat vaatimukset.

Vaatimusten viimeistely, jossa vaatimukset muotoillaan sovittuun esitysmuotoon ja järjestetään sopivaan järjestykseen.

Vaatimusten kelpuutus, jolla dokumentoidut vaatimukset hyväksytään sisältönsä ja laatunsa puolesta, tavallisimmin katselmoimalla.

Vaatimusten priorisointi, jossa vaatimukset asetetaan tärkeysjärjestykseen.

Vaatimusten dokumentointi, jossa tuotetaan vaatimusdokumentti (tai -tietokanta).

Vaatimusten muutosten hallinta, jolla hallitaan projektin tms. aikana vaatimuksiin tulevia muutoksia ja niiden vaikutuksia projektiin.

Perusprosessi muistuttaa ns. vesiputousmallia, jossa tehtävät seuraavat peräkkäin toisiaan. Käytännössä tehtävät eivät aina seuraa toisiaan edellä esitetyssä järjestyksessä, koska työn kuluessa tulee esille usein uutta tietoa, jonka vuoksi on palattava täydentämään edellisiä vaiheita, eli prosessiin kuuluu iteraatiokierroksia.

Tästä prosessista on käytössä monia eri variaatioita eri nimityksillä, mutta niissä on yleensä edellä esitetyt perustehtävät.

Spiraalimallissa (viite [3-9]) tehtävät toistuvat useassa jaksossa ja vaatimusjoukko kasvaa vähitellen kierros kierrokselta, kunnes vaatimusjoukko on riittävän hyvin määritelty.

REPEAT-malli (viite [3-9]) on vaatimusten tilaan perustuva malli, jossa vaatimusten on läpäistävä eri tilat, joihin liittyy määriteltyjä toimenpiteitä.

Viitteen [3-10] mukaan **vaatimusmäärittelyyn** kuuluvat seuraavat kolme osaprosessia: vaatimusten selvitys, määrittely ja kelpuutus.

Viitteessä [3-11] taas vaatimusmäärittely jaetaan viiteen osaprosessiin: vaatimusten selvitys, vaatimusten analyysi, vaatimusten määrittely, vaatimusten kelpuutus ja vaatimushallinta. Tämän mukaan vaatimusmäärittely onkin yläkäsite ja vaatimushallinta sen alakäsite.

Viiteen [3-12] vaatimusmäärittely taas koostuu seuraavista vaiheista: vaatimusten selvitys, vaatimusanalyysi, vaatimusneuvottelu ja vaatimusten dokumentointi.

Viitteessä [3-11] ohjelmistojen vaatimusmäärittelyyn taas katsotaan kuuluvaksi sovellusalaanalyysi, sidosryhmäanalyysi, tavoitteiden ja skenaarioiden määrittely, vaihtoehtojen tutkiminen, riskianalyysi, neuvottelu, vaihtoehtojen dokumentointi, määrittely, todentaminen, kelpuutus ja muutoshallinta.

Eri lähestymistavoissa vaatimusmäärittelyyn luetaan siis kuuluvaksi erilaisia osaprosesseja. Tosin eri osaprosessien sisältö voi vaihdella ja siten prosessi kokonaisuudessaan on keskeisimmiltä osiltaan melko samanlainen. Osaprosessijaolla haluttaneen lähinnä korostaa tiettyjen osavaiheiden merkitystä.

Lähteen [3-14] mukaan *ei ole olemassa yhtä ainoata vaatimusmäärittelyprosessia*, joka sopisi kaikille organisaatioille. Jokaisen organisaation on kehitettävä oma prosessinsa, joka sopii siinä kehitettäville järjestelmille, sen omaan organisaatiokulttuuriin sekä organisaatiossa työskentelevien ihmisten kokemukseen ja osaamiseen.

Tässä raportissa kahdesta vastakkaisesta ryhmittelyyn liittyvästä määrittelystä on valittu *vaatimushallinta yläkäsitteeksi ja vaatimusmäärittely sen osaksi*.

3.6.2 Vaatimushallinnan perusprosessi

Selvityksen perusteella voidaan vaatimushallintaan katsoa kuuluvan seuraavat osaprosessit:

1. Järjestelmäympäristöanalyysi
2. Vaatimusten selvittäminen
3. Vaatimusanalyysi ja -määrittely (ml. vaatimusten priorisointi)
4. Vaatimustietojen hallinta (ml. vaatimusten dokumentointi sekä jäljitettävyyden ja muutosten hallinta)

5. Vaatimusten katselmointi ja hyväksyminen
6. Vaatimusten käyttö.

3.6.2.1 Järjestelmäympäristöanalyysi

Järjestelmäympäristöanalyysissä selvitetään, mihin ongelmaan järjestelmällä haetaan ratkaisua ja mihin toimintaympäristöön se liittyy. Analyysin tuloksena saadaan mm. tarpeellista taustatietoa sekä kuvaus nyky- ja tavoitetilasta sekä hahmotelma järjestelmästä ja sen käyttäjistä sekä sen ajatellusta käytöstä ja käyttötavoista.

3.6.2.2 Vaatimusten selvittäminen (requirements elicitation)

Vaatimusten selvittäminen katsotaan usein ensimmäiseksi vaiheeksi vaatimusmäärittelyssä [3-14]. Jo selvitys-termi ilmaisee sen, että vaatimukset eivät ole jossain valmiina odottamassa, vaan niiden saamiseksi on tehtävä työtä. Selvityksen aikana kerätty tieto on vielä tulkittava, analysoitava, mallinnettava ja kelpuutettava ennen kuin voidaan katsoa, että riittävän laadukas ja kattava vaatimusjoukko on aikaansaatu.

Vaatimusten selvitystä ohjaavat pitkälti projektissa valittavat mallinnusmenetelmät, jotka määräävät, millaista vaatimusten selvitystekniikkaa on käytettävä.

Eräs vaatimusten selvittämisen tärkeitä tavoitteita on määrittää järjestelmän rajat, koska siitä riippuvat sitä seuraavat selvitystehtävät [3-14]. Sidosryhmien selvittäminen on myös kriittistä, koska järjestelmä tehdään sidosryhmiä varten. Järjestelmän tavoitteiden selvittäminen on myös oleellista, jotta ymmärretään paremmin, mihin ongelmiin järjestelmällä haetaan ratkaisuja.

Vaatimusten selvittämiseen liittyy myös vaatimusten tarkastus sidosryhmien toimesta silloin, kun ne ovat olleet mukana työssä. Tarkastuksessa varmistetaan, että sidosryhmätarpeiden tulkinta on tehty oikein.

3.6.2.3 Vaatimusten selvitystekniikoita

Käytettävä selvitystekniikka riippuu työhön käytettävissä olevasta ajasta ja muista resursseista sekä selvitettävän tiedon laadusta [3-14]. Erotettavissa on seuraavanlaisia selvitystekniikkaryhmiä:

- perinteiset tekniikat, kuten kyselyt, haastattelut, dokumenttien analysointi, organisaatiokaaviot, prosessimallit, standardit ja käyttöym. käsikirjat.
- ryhmäselvitystekniikat, kuten aivoriihet, fokusryhmät ja JAD-workshopit.
- prototyypit, joita käytetään erityisesti silloin, kun on olemassa suuri epävarmuus vaatimusten suhteen.
- mallipohjaiset tekniikat, joissa käytetään erityistä mallia tiedon keräämisen ohjaamisessa. Tällaisia ovat tavoitepohjaiset menetelmät (esim. KAOS) sekä skenaariopohjaiset menetelmät (esim. CREWS).
- kognitiiviset tekniikat on alunperin kehitetty tietämyspohjaisten järjestelmien kehittämiseen. Ne käsittävät esim. protokolla-, analyysi-, porrastus- (laddering), korttien lajittelusekä matriisitekniikoita.
- asiayhteystekniikat, kuten esimerkiksi etnografiset, etnometodologiset ja keskusteluanalyysitekniikat.

3.6.2.4 Vaatimusten selvitysprosessi

Selvitystekniikoiden käytön ohjaamiseksi tarvitaan ohjausta prosessin muodossa [3-14]. Menetelmät tarjoavat erään ohjaustuen. Kukin menetelmä omaa omat vahvuutensa ja heikkoutensa, ja on normaalisti parhaiten sopiva tietylle sovellusalueelle, esim. ohjelmistokehitykseen. Esimerkiksi CREWS [3-15] tarjoaa menetelmän vaatimusten selvittämiseen käyttäen apuna käyttötapauksia (use cases) ja skenaarioita.

Kaikissa tapauksissa ei ole tarpeen soveltaa täysmittaista menetelmää tai prosessia, vaan voidaan valita vain pelkkä tilanteeseen soveltuva tekniikka tai tekniikoita.

3.6.2.5 Vaatimusanalyysi ja -määrittely

Vaatimusanalyysissä jalostetaan selvitysvaiheessa tuotetusta aineistosta sidosryhmävaatimukset. Jalostuksessa käytetään apuna vaatimusmallinnusta.

Määrittelyvaiheessa tuotetut vaatimukset kirjoitetaan oikeaan vaatimusmuotoon sekä järjestetään ja luokitellaan ne käyttäen usein tähän tarkoitukseen suunniteltuja vaatimushallintatyökaluja.

Sidosryhmävaatimuksista johdetaan tämän jälkeen järjestelmävaatimukset käyttäen soveltuvia mallinnusmenetelmiä ja -tekniikoita apuna.

3.6.2.6 Vaatimustietojen hallinta

Vaatimustietojen hallintaan sisältyy vaatimustietojen dokumentointi, vaatimusten jäljitettävyyden luonti ja ylläpito sekä vaatimusten muutoksiin liittyvä tietojen hallinta.

Vaatimustietoja dokumentoidaan käyttäen toimisto-ohjelmistoja (esim. Word, Excel) tai erityisesti vaatimushallintaan tarkoitettuja työkaluja (esim. DOORS, Caliber). Suurten vaatimusmäärien hallitsemiseksi on syytä käyttää tarkoitukseen kehitettyjä vaatimushallintatyökaluja.

Jäljitettävyyteen liittyvät liitynnät (linkit) luodaan jo vaatimusmäärittelyvaiheessa ja niitä ylläpidetään samalla kuin muitakin vaatimustietoja. Jäljitettävyys edellyttää yleensä myös hyvän vaatimushallintatyökalun käyttöä.

Vaatimusten muutosten hallintaan kuuluu muutosten vaikutusten analysointi ja muutosten hyväksyminen. Ilman systemaattista muutosproseduuria ei vaatimuksia saa muuttaa kesken projektin.

3.6.2.7 Vaatimusten katselmointi ja hyväksyminen

Vaatimukset katselmoidaan niiden valmistuttua. Katselmoinnissa arvioidaan vaatimuksia sekä varmistetaan yksittäisten vaatimusten sekä koko vaatimusjoukon laatu. Katselmoidut vaatimukset hyväksytään suunnittelun jatkon pohjaksi katselmoinnin jälkeen.

3.6.2.8 Vaatimusten käyttö

Vaatimusten käyttöön sisältyy niiden käyttö suunnittelussa sekä niiden todentaminen ja kelpuutus.

Vaatimusten pohjalta kehitetään sitä seuraavan suunnitteluvaiheen vaatimukset tai ratkaisut. Esimerkiksi järjestelmävaatimusten pohjalta määritetään järjestelmän toiminnallinen ja fyysinen arkkitehtuuri.

Todentamisella osoitetaan, että kunkin suunnittelu- ja toteutusvaiheen tulokset täyttävät edeltävät vaatimukset. Esimerkiksi järjestelmävaatimukset todennetaan sidosryhmävaatimuksia vastaan ja suunnitteluratkaisut järjestelmävaatimuksia vastaan.

Kelpuutuksella osoitetaan, että valmis järjestelmä täyttää sidosryhmävaatimukset. Kelpuutus tapahtuu valmiilla järjestelmällä oikeassa käyttöympäristössä ja -olosuhteissa.

3.7 Vaatimushallinnan State–of–the–Art-käytäntöjä

3.7.1 Yleistä

Tässä tehdyn selvityksen perusteella kehittyneimmät saatavilla olevat vaatimushallintakäytännöt (State–of–the–Art) ovat yleispäteviä ja sopivat yleensä käytettäväksi useilla sovellusalueilla. Pelkästään ohjelmistojen tuottamiseen on kuitenkin kehitetty nimenomaan niihin soveltuvia vaatimushallintakäytäntöjä, jotka eivät sellaisenaan sovellu vain laitteistoja (hardware) sisältävien järjestelmien vaatimushallintaan. Tällaisia ovat mm. monet muodolliset vaatimusten kuvauskielet.

Tietty sovellusalue, -tehtävä tai -ympäristö voi toisaalta tarvita erityisen fokusoinnin tai erityisen tekniikan vaatimusmäärittelyssä [3-16]. Tämä pätee esimerkiksi reaktiivisten säätöjärjestelmien tapauksessa, joille on kehitetty mm. SCR-menetelmä [3-17, 3-18] (Software Cost Reduction) ja CORE-menetelmä [3-19].

3.7.2 Turvallisuuskriittisten järjestelmien vaatimusmäärittely

Turvallisuuskriittisiltä järjestelmiltä, kuten esimerkiksi ydinvoimalaitosten digitaalisilta I&C-järjestelmiltä, edellytetään erityisen korkeaa turvallisuustasoa [3-20]. Vaatimusmäärittelyt näyttelevät merkittävää roolia tällaisten järjestelmien turvallisuuden arvioinnissa. Vaatimusdokumenttien katselmoinnit ja testit sekä jäljitettävyyden hallinta nähdään tärkeiksi keinoiksi turvallisuuden varmistamisessa. Lisäksi osapuolten välisen kommunikaation on toimittava. Kaikki nämä muodostavat tärkeän perustan esimerkiksi automaatiojärjestelmien ja ohjelmoitavien laitteiden suunnittelussa ja toteutuksessa.

Edellisten tekijöiden lisäksi on vaatimusmäärittelyjen laadulla selvä vaikutus lopputulokseen. Viitteessä [3-21] on esitetty yhteenvetoa muodollisten määrittelyjen (formal specifications) soveltuvuudesta uusien järjestelmien kehittämiseen ja olemassa olevien järjestelmien parantamiseen. Sen mukaisesti muodollisten määrittelyjen avulla saatiin korkealaatuisia tuloksia mm. liikenne-, tieto-, tietoliikenne- ja voimalaitosten säätö- ja turvallisuusjärjestelmien kehittämisessä.

Viitteessä [3-22] on esitetty turvallisuuskriit-

tisten ohjelmistojen kehittämiseen liittyviä menettelyjä ml. vaatimusmäärittelyyn liittyviä tekijöitä. Siinä nähdään muodolliset vaatimusmäärittelyt ja vaatimusten analysointi sekä kelpuutus tärkeiksi keinoiksi ohjelmiston laadun varmistamisessa.

3.7.3 Vaatimusmäärittelyn lähestymistavat

Seuraavassa on esitetty selvityksessä tunnistetut vaatimushallintalähestymistavat.

Kartoituksessa on tunnistettu nykyään yleisesti käytettyjä vaatimusmäärittelyn lähestymistapoja. Niiden erilaisia sovelluksia löytyy maailmalta lukuisasti. Keskeisimmät lähestymistavat ovat seuraavat:

1. Toimintopohjainen vaatimusmäärittely (function-oriented)
2. Tavoitepohjainen vaatimusmäärittely (goal-oriented)
3. Skenaariopohjainen vaatimusmäärittely (scenario-oriented)
4. Oliopohjainen vaatimusmäärittely (object-oriented)
5. Agenttipohjainen vaatimusmäärittely (agent-oriented)
6. Näkökulmapohjainen (viewpoint) vaatimusmäärittely
7. Muita lähestymistapoja.

3.7.3.1 Toimintopohjainen vaatimusmäärittely

Toimintopohjaista vaatimusmäärittelyä edustaa esimerkiksi SADT [3-23] ja sen sovellukset. SADT:ssa keskeisenä on datapohjainen lähestymistapa ja tärkeimpänä esitysmuotona ovat datavuokaaviot.

Toiminnallinen vaatimusmäärittely painottaa järjestelmän toiminnallista osittamista ja pääpaino on järjestelmän käyttäjille tarjoamissa palveluissa. Järjestelmä ja vaatimukset on organisoitu näiden toimintojen ympärille.

Toiminnallisen analyysin avulla tuotetaan tavallisesti toimintolohkokaavio tai muu vastaava esitys järjestelmän toiminnoista. Toiminnot jaetaan osatoimintoihin ja järjestetään hierarkkisesti toimintopuuksi tai -lohkokaavioksi. Tämä hierarkia toimii alussa vaatimusten järjestämisen rakenteena, mutta sitä voi suunnitteluprosessin kuluessa joutua muuttamaan. Toimintojen perusteella tuotetaan niihin liittyviä toiminnallisia vaatimuksia. Toiminnallisiin vaatimuksiin liittyvät sekä yleiset ei-toiminnalliset vaatimukset määritetään erikseen.

Toiminnallista mallinnusta voidaan käyttää sekä sidosryhmä- että järjestelmävaatimusten tuottamisen, järjestämisen ja todentamisen apuna.

3.7.3.2 Tavoitepohjainen vaatimusmäärittely

Tavoitepohjainen vaatimusmäärittely [3-24, 3-5, 3-25] pohjautuu järjestelmän ylätason visioon ja tavoitteisiin. Tavoitteet tukevat vaatimusmäärittelyä niiden kehittämisessä, todentamisessa, kelpuutuksessa, ristiriitojen ratkaisussa ja neuvotteluissa sekä vaatimusten perusteluissa ja muutoksissa.

Tavoitteet määritetään ja mallinnetaan prosessin alussa. Mallinnukseen kuuluu myös tavoitteiden luokittelu sekä tavoitteiden attribuuttien sekä liityntöjen (linkkien) määrittely.

Tavoitteiden määrittely voidaan tehdä käyttäen erilaisia kuvaustapoja epämuodollisesta muodolliseen. Mallinnus ja määrittely toimivat tukena vaatimusmäärittelyn eri vaiheissa kuten vaatimusten kehittämisessä, johdonmukaisuuden ja kattavuuden tarkastuksessa, vaihtoehtojen valinnassa, vaatimusten muutosten hallinnassa jne.

Tavoitepohjaisen VM:n avulla voidaan myös todentaa, että kehitetyt vaatimukset täyttävät tunnistetut tavoitteet. Tämä voidaan tehdä joko epämuodollisen tai muodollisen tarkastelun avulla.

Tavoitteet voidaan hyväksyä (kelpuuttaa) skenaarioiden avulla. Skenaarioiden käyttöä tähän on kuvattu esim. lähteessä [3-26].

Osatavoitteiden ja vaatimusten kehittäminen voidaan tehdä esim. Miten-kysymyksillä [3-26] ja [3-27], joilla saadaan kehitettyä vaatimus–tavoitehierarkiaa alaspäin. Miksi-kysymyksillä taas saadaan kehitettyä hierarkiaa ylöspäin. Tavoitteita voidaan johtaa myös kehittyneemmillä menetelmillä, joihin on viitattu lähteessä [3-24].

Tavoitepohjaista menetelmää soveltaa mm. KAOS-menetelmä, jota on myös kuvattu lähteessä [3-27]. KAOS-menetelmässä käytetään sen omaa muodollista kuvauskieltä ja tietämyspohjaa. KAOS-menetelmää on lähteen [3-27] mukaan käytetty 11 projektissa, mm. lennonvalvontajärjestelmän sekä sairaalan ensiapujärjestelmän kehittämisessä.

Muita tavoitepohjaisia menetelmiä tukevat mm. Albert-kuvauskieli [3-28], i*- [3-28], NFR- [3-29] ja NATURE-lähestymistavat [3-30].

3.7.3.3 Skenaariopohjainen vaatimusmäärittely

CREWS-tutkimuksessa [3-31] kehitettiin lähestymistapaa ja työkalu skenaarioiden käytölle vaatimushallinnassa.

Skenaariot kuvaavat joko tekstimuotoisena tai graafisesti peräkkäisiä vaiheita tai askelia, jonka nykyinen tai suunniteltava järjestelmä suorittaa. Sen tarkoitus on stimuloida ja dokumentoida ajatuksia olemassa olevista ongelmista, tapahtumista ja niihin liittyvistä oletuksista sekä toimintamahdollisuuksista ja riskeistä. Skenaariot tarjoavat keskitason abstraktion mallien ja todellisuuden välillä, jotka toimivat maailmanlaajuisesti ymmärrettynä, osallistuvan suunnittelun välineenä sekä helpottavat suunnittelutietämyksen uudelleenkäyttöä.

Vaatimusten selvityksen alkuvaiheessa skenaariot kohdentuvat nykyisen järjestelmän ongelmiin. Siten ne helpottavat muutostavoitteiden tunnistamista ja auttavat johtamaan yksityiskohtaisempia tavoitteita. Kun tulevan järjestelmän vaatimukset on määritelty, voidaan tuottaa skenaarioita vaatimusten kelpuuttamiseen todellisuutta ja korkean tason tavoitteita vastaan, mutta myös auttamaan jalostamaan vaatimuksia esimerkiksi poikkeustilanteiden käsittelyyn.

Viitteessä [3-32] on kuvattu selvitystä skenaariopohjaisen lähestymistavan käytöstä 15 projektissa eri Euroopan maissa. Selvitys osoittaa, että skenaarioita käytetään hyvin monella eri tavalla ja että tarvitaan enemmän menetelmällistä ohjausta sekä parempaa työkalutukea skenaariopohjaisessa vaatimusmallinnuksessa.

3.7.3.4 Oliopohjainen vaatimusmäärittely

Viitteessä [3-33] kuvataan kehitettyä oliopohjaista lähestymistapaa. Oliopohjainen mallinnus on yleisesti käytetty menetelmä ohjelmistojen tuotannossa. Vaatimusmallinnuksessa järjestelmää tarkastellaan olioiden (objektien) ja niiden ominaisuuksien avulla.

Oliopohjaista mallia on viitteen [3-33] mukaan helppo muuttaa ja laajentaa projektin kuluessa. Mallinnus verrattuna toiminnalliseen mallinnukseen on myös helppoa mallin kuvauksen ollessa lähellä fyysistä järjestelmää; olioiden uudelleenkäyttömahdollisuus on parempi ja suunnitteluprosessi voidaan nähdä pelkästään oliomallin kehittymisenä radikaalien muutosten sijasta; mallin kelpuutus sidosryhmien toimesta on myös helpompaa.

3.7.3.5 Agenttipohjainen vaatimusmäärittely

Agentti on itsenäinen olio tai elementti (ihminen, ohjelmisto, laite, järjestelmä tms.), joka on vuorovaikutuksessa ympäristönsä kanssa; ne ovat riippumattomia, ne kommunikoivat toisten agenttien kanssa, reagoivat ”ärsykkeisiin”, ne ovat liikkuvia, ennakoivia, älykkäitä, niillä on ”luonnetta”, ne oppivat ja kehittyvät.

Viitteissä [3-34, 3-35] on kuvattu agenttipohjaista lähestymistapaa vaatimusmallinnukseen ja käsitelty myös monia muita lähestymistapoja, joissa on samanlaisia piirteitä kuin kaavaillussa agenttilähestymistavassa. Lähestymistapa ei ole aivan samanlainen kuin mikä on käytössä ohjelmistotuotannossa. Menetelmä ei vielä ole kovin pitkälle kehitetty, mutta dokumentin mukaan se voisi tarjota paremman vaihtoehdon vaatimusmäärittelyyn verrattuna moniin muihin lähestymistapoihin. Viitteessä [3-35] on esitetty agenttilähestymistapaa soveltavista erityismenetelmistä CSD- (Composite systems Design), KAOS-lähestymistavat ja F3- ja i*-viitekehys sekä Albert II-kuvauskieli.

Viitteessä [3-25] on kuvattu yhdistelmää, jossa on sovellettu tavoite- ja agenttipohjaista lähestymistapaa valmiista komponenteista rakennettavien järjestelmien vaatimusmäärittelyyn (CARE-menetelmä). Yleensä suurin osa vaatimusmäärittelymenetelmistä olettaa, että järjestelmä luodaan aivan tyhjästä, kun tilanne useimmiten on se, että järjestelmä kootaan osista. Dokumentti osoittaa tällaisen järjestelmäkehityksen omat erityispiirteet ja ongelmat sekä kuvatun menetelmän keinot niiden ratkaisemiseksi.

3.7.3.6 Näkökulmapohjainen vaatimusmäärittely

Näkökulmapohjaista lähestymistapaa kehitettiin jo 1980-luvulla, mitä edustaa esimerkiksi CORE-menetelmä [3-36]. Uudempi Viewpoint-malli [3-23] perustuu erilaisten näkökulmien avulla tapahtuvaan tarkasteluun ja sitä kautta vaatimusten tunnistamiseen. Näkökulmia suositellaan valittavaksi vähintään kolme, mutta ei mielellään yli kuutta.

Perustelut näkökulmapohjaisen lähestymistavan puolesta ovat:

1. Järjestelmän käyttö on heterogeenista – ei ole sellaista kuin tyypillinen käyttäjä. Näkökulmat organisoivat eri käyttäjien ja sidosryhmien vaatimuksia.
2. Erilaista tietoa sovellusalueelta, ympäristöstä ja järjestelmän kehittämisestä tarvitaan järjestelmän määrittelyyn. Näkökulmia voidaan käyttää tämän tiedon keräämiseen ja järjestämiseen.
3. Näkökulmia voidaan käyttää apuna vaatimusten selvittämisprosessin strukturoinnissa.
4. Näkökulmia voidaan käyttää runkona erilaisille järjestelmämalleille, jotka tarjoavat järjestelmän määrittelytietoa.
5. Näkökulmia voidaan käyttää strukturoimaan vaatimuskuvauksia ja paljastamaan ristiriitoja vaatimusten välillä.

Näkökulmat voidaan luokitella kolmeen pääryhmään: vuorovaikutus-näkökulmaan (interactor), epäsuoraan sidosryhmänäkökulmaan ja sovellusalue-näkökulmaan:

- vuorovaikutus-näkökulma tarkastelee järjestelmän ja sen kanssa vuorovaikutuksessa olevan tekijän (järjestelmä, kone tai ihminen) välistä toimintaa (esim. käyttäjä, lentäjä, lentokone).
- epäsuoran sidosryhmän näkökulma ottaa huomioon sellaiset sidosryhmät, jotka eivät suoranaisesti ole tekemisissä järjestelmän kanssa, mutta joilla on jokin intressi järjestelmän suhteen (esim. kunnossapitäjä tai viranomainen).
- sovellusala-näkökulma käsittelee alan sisäisiä erikoistietoja tai -osaamista, joka on otettava huomioon järjestelmän kehittämisessä (esim. sähkömagneettinen säteily-ympäristö).

Näkökulmat-lähestymistapa auttaa parantamaan vaatimusmäärittelyn laatua sekä vaatimusmäärittelyssä että vaatimusten järjestämisessä. Se soveltuu käytettäväksi yhdessä muiden lähestymistapojen kanssa ja siinä voidaan käyttää erilaisia vaatimusten kuvaustapoja.

Viewpoint-Oriented Requirements Definition, VORD [3-37] soveltaa myös näkökulmalähestymistapaa. Se ottaa huomioon sekä loppukäyttäjän että organisaation näkemykset. VORD-työkalu on myös kehitetty.

3.7.4 Muita vaatimusmäärittelyn lähestymistapoja

3.7.4.1 QFD vaatimusmäärittelyssä

QFD (Quality Function Deployment) on jo 1960-luvulla kehitetty sekä erityisesti 1990-luvulla laajempaan käyttöön levinnyt menetelmä tuotteiden tai palveluiden suunnitteluun, jota voidaan käyttää myös vaatimusmäärittelyyn [3-38]. Menetelmän avulla voidaan muuntaa ”asiakkaan ääni” eli asiakastarpeet mitattaviksi teknisiksi vaatimuksiksi ja suunnittelutavoitteiksi.

Keskeinen työkalu QFD:ssä on ”laatutalo”, mikä muodostaa viisivaiheisen prosessin:

- Vaiheessa 1 tunnistetaan asiakastarpeet.
- Toisessa vaiheessa vaatimusinsinööri tuottaa asiakastarpeiden pohjalta tekniset vaatimukset ja niiden tavoitearvot. Vaatimukset kuvaavat, miten järjestelmä saavuttaa yhden tai useamman asiakastarpeista.
- Kolmannessa vaiheessa tunnistetaan teknisten vaatimusten ja asiakastarpeiden korrelaatiot ja niiden voimakkuudet. Ne tekniset vaatimukset, jotka eivät täytä mitään asiakastarvetta, voidaan hylätä. Toisaalta jokaisen asiakastarpeen on korreloitava ainakin yhden teknisen vaatimuksen kanssa.
- Neljännessä vaiheessa priorisoidaan asiakastarpeet. Tämä voidaan tehdä käyttäen esim. AHP-menetelmää (Analytic Hierarchy Process). Priorisoinnissa otetaan huomioon myös mahdolliset kilpailijoiden tuotteet.
- Vaihe 5 täydentää prosessin priorisoimalla tekniset vaatimukset ottaen huomioon asiakastarveprioriteetit ja niiden korrelaatiot teknisiin vaatimuksiin. QFD:ssä kaikki tekniset vaatimukset ovat jäljitettävissä asiakkaan tarveilmaisuihin asti.

QFD:ssä on monia samanlaisia piirteitä kuin muissakin vaatimusmäärittelymenetelmissä. Se, kuten muutkin menetelmät, ei ole täydellinen. Hyvä vaatimusinsinööri soveltaa useampaa menetelmää, prosessia ja tekniikkaa, jotka parhaiten soveltuvat kohteeseen.

3.7.4.2 REVEAL-malli

REVEAL-malli [3-39] pohjautuu Michale Jacksonin kehittämään systemaattiseen menetelmään, josta Praxis Critical Systems on kehittänyt REVEAL-mallin. Se yhdistää tuoreen tutkimustiedon

ja tekniikat käytännölliseksi ja selkeäksi menetelmäksi vaatimusten selvittämiseen ja määrittelyyn, määrittelyn selkeyden parantamiseen ja vaatimusten muutosten hallintaan. Se soveltuu mm. turvallisuuskriittisten järjestelmien vaatimusmäärittelyyn.

REVEAL-malli koostuu neljästä päävaiheesta sekä sen jälkeisestä vaatimusten ylläpitovaiheesta. Se alkaa ongelman asiayhteyden (context) määrittämisestä, jatkuu sidosryhmien tunnistamisella, vaatimusten selvittämisellä, tallentamisella, todentamisella ja kelpuutuksella sekä jatkuen edelleen vaatimusten käytöllä ja ylläpidolla. REVEAL ei vaadi minkään erityisen kuvaustavan tai työkalun käyttöä. Mallia voidaan siis soveltaa yleisesti käytettävissä olevilla vaatimushallintatyökaluilla, esim. DOORS tai Cradle.

3.7.4.3 Laaja vaatimusmäärittelymalli

Laaja vaatimusmäärittelymalli [3-40] on tietojärjestelmien määrittelyyn kehitetty malli. NATURE-projektissa kehitetty vaatimusmäärittelyprosessi on päätösorientoitunut. Siinä lähtökohtana on visio järjestelmän tarpeesta. Visio jaetaan tavoitteiksi ja osatavoitteiksi käyttäen osittamismenetelmiä. Tämän osittamisprosessin yhteys strukturoidaan käyttäen tietojärjestelmäsuunnittelun ”neljää maailmaa”: sisältö- (subject), käyttö- (usage), järjestelmä- (system) ja kehitysmaailma (development). Prosessissa käytetään lisäksi vaatimusmäärittelyn kolmea dimensiota: määrittely-, hyväksymis- ja esitysdimensioita. Osatavoitteet jaetaan edelleen osiin, kunnes on saavutettu taso, jolla perusratkaisustrategiat on löydettävissä.

3.7.4.4 Software cost reduction (scr)

SCR [3-41] on kehitetty suurten, reaaliaikaisten sulautettujen järjestelmien määrittelyyn.

3.7.4.5 Volere-template

Volere-template [3-42] on vaatimusmäärittelymalli, joka pitää sisällään vaatimuksiin liittyvät tiedot sekä niiden tuottamiseen liittyvän prosessin kuvauksen sekä vaatimushallintaan liittyvää tietoa. Mallia voidaan soveltaa useilla eri työkaluilla, esim. Requisite, DOORS ja Caliber.

3.7.4.6 Muita vaatimusmäärittelymenetelmiä

Viitteessä [3-43] on vertailtu kahdeksaa vaatimusmäärittelymenetelmää niiden soveltuvuuden kannalta. Vertaillut menetelmät ovat BSM, OCTOPUS, ROOM, SA/RT, SCR, SDL, UML JA Z. Osa näistä on vain kuvauskieliä (SCR, SDL, UML JA Z) eivätkä ne sisällä varsinaista menetelmää vaatimusten määrittämiseksi. Menetelmät on jaettu toiminto- (BSM, SCR, SA/RT JA Z) ja oliopohjaisiksi (OCTOPUS, ROOM, SDL JA UML). Vertailun päälöydös on, että menetelmässä käytetty vaatimusten strukturointitapa (esim. hierarkia ROOM:ssa tai näkymä UML:ssä) on suoraan verrannollinen vaatimusmäärittelyssä esiintyviin ongelmiin.

Viitteessä [3-44] on kuvattu yksinkertaista 5WH-menettelyä (What, Who, Where, When, Why, How) vaatimusten tunnistamiseksi. Se auttaa muuntamaan sidosryhmien tarpeet, halut ja odotukset laadukkaaksi ja kattavaksi vaatimusjoukoksi.

Viitteessä [3-45] on esitetty menetelmä, jossa on yhdistetty oliopohjainen ja SDL-lähestymistapa (SDL = Specification and Description Language). Sen tavoitteena oli parantaa järjestelmän suunnitteluprosessia painottaen vaatimusanalyysiä ja suunnittelua. Menetelmää on sovellettu tietoliikennepalvelun (ääniposti) suunnittelussa.

3.7.5 Vaatimusten kuvaustavat

3.7.5.1 Yleistä

Vaatimuksen kuvaus siten, että se sisältää tarvittavan informaation, edellyttää usein useammanlaisen kuvaustavan käyttöä. Vaatimus voidaan kuvata luonnollisen kielen muotoisena tekstinä, graafisena esityksenä tai jonkin muodollisen kuvauskielen avulla.

Yleisesti erityyppiset kuvaustavat sopivat eri tavalla eri elinkaaren vaiheisiin. Esimerkiksi alkuvaiheen vaatimusten kuvaustavaksi sopivat parhaiten epämuodolliset tavat (useimmiten luonnollisella kielellä ilmaistuna), kun taas suunnittelun myöhempään vaiheeseen sopivat puolimuodolliset ja muodolliset kuvaustavat. Turvallisuus- tai muissa kriittisissä järjestelmissä tai järjestelmän osissa on tarpeen muodollisten kuvaustapojen käyttö luotettavan lopputuloksen varmistamiseksi.

3.7.5.2 Vaatimusten kuvaustekniikoita

Luonnollisen kielen avulla vaatimuksia kuvataan kaikkein yleisimmin.

Viitteessä [3-1] on esitetty vaatimusten mallinnuksen ja määrittelyn erilaisia tekniikoita, jotka on jaettu puolimuodollisiin (semi-formal) ja muodollisiin (formal) tekniikoihin.

Puolimuodollisia tekniikoita ovat mm.
- kohdemallit (entity-relationship diagram)
- tila-muutos-diagrammit (state-transition diagram)
- datavuokaaviot (data flow diagram).

Muodollisia tekniikoita ovat mm.
- historiapohjaiset määrittelyt: järjestelmän käyttäytyminen ajan funktiona
- tilapohjaiset määrittelyt: järjestelmän tilojen avulla
- muutostilapohjaiset määrittelyt: kuvaus tilamuutosten avulla
- funktiomäärittelyt: järjestelmän kuvaus matemaattisten funktioiden avulla
- toimintamäärittelyt (operational): järjestelmä on kokoelma prosesseja.

Edellä esitettyjen tekniikoiden puutteita on esitetty viitteessä [3-1] , joista tärkeimmät ovat:
- ne eivät käsittele MIKSI- ja KUKA/MIKÄ-aspekteja vaan ainoastaan MITÄ-aspektia,
- ne eivät mahdollista vaihtoehtoisten ratkaisujen esittämistä eivätkä ei-toiminnallisia näkökohtia,
- ne eivät erota esim. vaatimuksia ja ohjelmistomäärittelyä eivätkä mahdollista puolimuodollisten ja muodollisten kuvausten yhdistelmiä eri kohdissa.

Em. kuvaustekniikoiden puutteiden korjaamiseksi on kehitetty esim. KAOS-menetelmään pohjautuva tavoitepohjainen lähestymistapa.

3.8 Vaatimustenhallintakäytännöt, yhteenveto

Selvityksen perusteella voidaan todeta, että
- käytössä on hyvin laaja joukko erilaisia käytäntöjä
- käytäntöjen pääpaino on ohjelmistotuotannon vaatimushallinnassa
- toiminnallisten vaatimusten määrittelyyn on enemmän käytäntöjä kuin ei-toiminnallisille vaatimuksille
- selkeää kuvaa jonkin menetelmän soveltuvuudesta tietylle sovellusalueelle ei selvityksen perusteella saa
- soveltuvuutta eri sovellusalueille voisi selvittää perehtymällä perusteellisemmin case-esimerkkitapauksiin
- useimmat käytännöt soveltuvat monelle sovellusalueelle
- tuloksissa on esitelty käytäntöjä, jotka nousseet esille tietohaun yhteydessä ja jotka näyttävät tulevan esille useassa yhteydessä
- jatkoselvitys on tarpeen kiinnostavimpien käytäntöjen soveltuvuuden selvittämiseksi paremmin tietylle sovellusalueelle
- tavoitepohjainen lähestymistapa on ominaisuuksiensa puolesta hyvä varmistamaan hyvän vaatimusmäärittelyn laadun
- muodolliset vaatimusmäärittelytavat, vaatimusten katselmoinnit sekä vaatimusten jäljitettävyyden hallinta ovat hyviä keinoja ohjelmistojen ja ohjelmistopainotteisten järjestelmien luotettavuuden varmistamisessa.

Osa 3 – Viiteluettelo

3-1 Axel van Lamsweerde, Building Formal Requirements Models for Reliable Software, http://www.info.ucl.ac.be/recherches/publications/publications/archives.html, ftp://info.ucl.ac.be/pub/publi/2001/avl-AdaEurope.pdf

3-2 Klaus Pohl, Requirements Engineering: An Overview, ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-96-02.pdf

3-3 U Nikula, J Sajaniemi, H Kälviäinen, Vaatimusten määrittely ja hallinta, http://cs.joensuu.fi/pages/saja/tSoft/dokumentit/20010515_nikula.pdf

3-4 Orlena C. Z. Gotel & Anthony C. W. Finkelstein, An Analysis of the Requirements Traceability Problem, http://citeseer.nj.nec.com/16558.html

3-5 Axel van Lamsweerde, Requirements engineering in the year 00: a research perspective (2000), Proceedings of the 22nd international conference on Software engineering, p. 5–19, June 04–11, 2000, Limerick, Ireland, http://citeseer.nj.nec.com/vanlamsweerde00requirements.html

3-6 Bashar Nusebeh, Steve Easterbrook, Requirements Engineering: A Roadmap, http://www.doc.ic.ac.uk/~ban/pubs/sotar.re.pdf

3-7 Gerald Kotonya and Ian Sommerville, Requirements Engineering With Viewpoints (1996), http://citeseer.nj.nec.com/kotonya96requirements.html

3-8 Colette Rolland, Modeling the Requirements Engineering Process, http://citeseer.nj.nec.com/rolland93modeling.html

3-9 Kotonya, G., Sommerville, I., (1997), Requirements engineering processes and techniques, John Wiley & Sons, ISBN: 0-471-97208-8, April 1998.

3-10 SWEBOK Trial version 1.0, http://www.swebok.org.

3-11 Dirk Aucher, Johan Blom, Roland Bol, Requirements Engineering in a Telecommunication Environment, http://www.docs.uu.se/astec/Reports/Reports/9710.ps.gz

3-12 Daniela E. Herlea Damian, Challenges in Requirements Engineering, Technical Report 99/645/08, 1999, University of Calgary, Canada, http://pharos.cpsc.ucalgary.ca/Dienst/Repository/2.0/Body/ncstrl.ucalgary_cs/1999-645-08/pdf

3-13 Linda Dawson, Paul Swatman, The Use of Object-Oriented Models in Requirements Engineering: A Field Study, http://aisel.isworld.org/password.asp?Vpath=ICIS/1999&PDFpath=crp9923.pdf

3-14 Bashar Nusebeh, Steve Easterbrook, Requirements Engineering: A Roadmap, http://www.doc.ic.ac.uk/~ban/pubs/sotar.re.pdf

3-15 Klaus Pohl, ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-96-02.pdf

3-16 Axel van Lamsweerde, Requirements engineering in the year 00: a research perspective (2000), Proceedings of the 22nd international conference on Software engineering, p. 5–19, June 04–11, 2000, Limerick, Ireland, http://citeseer.nj.nec.com/vanlamsweerde00requirements.html

3-17 K.L. Heninger, "Specifying Software Requirements for Complex Systems: New Techniques and their Application" (1980), IEEE Transactions on Software Engineering Vol. 6 No. 1, January 1980, 2–13, http://www.cs.virginia.edu/~cs340/materials/papers/heninger.pdf.

3-18 Constance Heitmeyer, SCR: A Practical Method for Requirements Specification, http://chacs.nrl.navy.mil/publications/CHACS/1998/1998heitmeyer-DASC98.pdf.

3-19 S. Faulk, J. Brackett, P. Ward and J. Kirby, "The CORE Method for Real-Time Requirements", IEEE Software, September 1992, 22–33, http://www.computer.org/software/so1992/s5022abs.htm

3-20 Terje Sivertsen, Rune Fredriksen and Atoosa P-J. Thunem, Jan-Erik Holmberg, Janne Valkonen, Olli Ventä ,Traceability and Communication of Requirements in Digital I&C Systems Development, (NKS-R project number NKS_R_2002_16) – Preproject Report

3-21 D. Graigen, S. Gerhart and T. Ralston, An International Survey of Industrial Applications of Formal Methods, Us Dept. Commerce, NIST, Computer Systems Lab., NISTGCR 93/626, March 1993, http://www2.umassd.edu/SWPI/FormalMethods/vol1.pdf.

3-22 Patrick R.H. Place, Kyo C. Kang, Safety-Critical Software: Status Report and Annotated Bibliography, Technical Report, CMU/SEI-92-TR-5, ESC-TR-93-182, http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr05.93.pdf

3-23 Sommerville, Sawyer, Viewpoints: Principles, Problems and Practical Approach to Requirements Engineering, http://www.comp.lancs.ac.uk/computing/users/cosh/AnnalsOfSE.pdf

3-24 Axel van Lamsweerde, Goal-Oriented Requirements Engineering: A Guided Tour (2001), Proceedings, Fifth IEEE International Symposium on Requirements Engineering (RE'01), Toronto, Canada, August 27–31, 2001. Pages: 249–262, http://citeseer.nj.nec.com/vanlamsweerde01goaloriented.html

3-25 Lawrence Chung and Kendra Cooper, A COTS-Aware Requirements Engineering Process: a Goal- and Agent-oriented Approach, http://www.utdallas.edu/~kcooper/research/INCOSE_2002_CARE.pdf.

3-26 A. van Lamsweerde, R. Darimont, and Ph. Massonet, "Goal-Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt (1995)", Proc. RE'95 – 2nd Intl. IEEE Symp. on Requirements Engineering, March 1995, 194–203, http://citeseer.nj.nec.com/vanlamsweerde95goaldirected.html

3-27 Goal-Driven Requirements Engineering: the KAOS Approach, http://www.info.ucl.ac.be/research/projects/AVL/ReqEng.html

3-28 Xiyun Wang, Yves Lesperance, Agent Oriented Requirements Engineering Using ConGolog and i*, http://www.cs.yorku.ca/~lesperan/papers/AOIS01.pdf

3-29 Luiz Marcio Cysneiros, Julio Cesar Sampaio do Prado Leite and Jaime de Melo Sabat Neto, A Framework for Integrating Non-Functional Requirements into Conceptual Models, http://citeseer.nj.nec.com/550290.html

3-30 NATURE report series, http://www-i5.informatik.rwth-aachen.de/PROJEKTE/NATURE/nature-reps-english.html

3-31 CREWS, http://panoramix.univ-paris1.fr/CRINFO/CREWS/Corps.htm

3-32 Klaus Weidenhaupt, Klaus Pohl, Matthias jarke, Peter Haumer, Scenario Usage in System Development: A Report on Current Practice, ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-97-16.pdf

3-33 Annie I. Antón, Thomas A. Gale, W. Michael McCracken and John J. Shilling, Object-Based Requirements Modeling for Process Continuity,' Proc. Twenty-Sixth Hawaii International Conference on System Sciences, Vol 3, pp. 191–202, 1993, http://www.csc.ncsu.edu/faculty/anton/publications.html

3-34 Eric Yu, Agent orientation as a Modelling Paradigm, http://www.cs.toronto.edu/~eric/.

3-35 Eric S. K. Yu, Why Agent-Oriented Requirements Engineering, http://www.cs.toronto.edu/pub/eric/REFSQ97.html

3-36 Systems Designers (1986); CORE – the manual; Internal Publication, SD-Scicon, http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/tara.pdf

3-37 João Araújo, Paulo Coutinho, Identifying Aspectual Use Cases Using a Viewpoint-Oriented Requirements Method, http://www.cs.bilkent.edu.tr/AOSD-EarlyAspects/Papers/AraujoCoutinho.pdf

3-38 Quality Function Deployment for Requirements Engineering, http://www.guydavis.ca/seng/seng613/group/qfd.shtml

3-39 REVEAL: A Keystone of Modern Systems Engineering, Praxis Critical Systems Limited 2000, reveal@praxis-cs.co.uk

3-40 Jarke M., Pohl K.; Establishing Visions in Context: Towards a Model of Requirements Processes, ftp://ftp.informatik.rwth-aachen.de/pub/NATURE/NATURE-93-10.ps.Z

3-41 Heitmeyer, Constance L., "Software Cost Reduction," Encyclopedia of Software Engineering, Two Volumes, John J. Marciniak, editor, ISBN: 0-471-02895-9, January 2002, http://chacs.nrl.navy.mil/personnel/heitmeyer.html

3-42 Volere Requirements Specification Template, http://www.systemsguild.com/GuildSite/Robs/Template.html

3-43 Antje von Knethen, Erik Kamsties, Ralf Reussner, Christian Bunse, Bin Shen, A Comparative Case Study with Industrial Requirements Engineering Methods, http://wwwagse.informatik.uni-kl.de/publications/paper/sea98.pdf.

3-44 James R van Gaasbeek, James N Martin, Getting to Requirements: the W5H challenge, http://www.incose-wma.org/docs/chapter/Jun02Centrevillepgm.pdf

3-45 Dirk Auchter, From Requirements Capture to esign: Combining the ARENA and SOMT method, http://www.docs.uu.se/astec/Reports/Reports/9713.ps.gz

4 Vaatimushallinnan soveltamismahdollisuudet

Työn tavoitteena oli tunnistaa vaatimushallinnan soveltamismahdollisuuksia viranomaistoiminnassa sekä voimayhtiöissä.

4.1 Vaatimushallinta viranomaistoiminnassa

Vaatimushallintaa voidaan soveltaa Säteilyturvakeskuksessa (STUK):

1. Uuden ydinvoimalaitoksen (FIN5) sekä toiminnassa olevien ydinvoimalaitosten valvontatoiminnan yhteydessä
   - YVL-ohjeiden ja muiden vastaavien ohjeiden täyttymisen valvonnassa: YVL-ohjeet sisältävät vaatimuksia, joiden täyttymistä todennetaan valvonnalla.
   - Asiakkaan (esim. ydinvoimayhtiö) laatujärjestelmän todentamisessa, mikä on ydinturvallisuusvalvontatoiminnan yksi osa-alue. Perustana todentamisessa ovat YVL-ohjeet, joissa on vaatimuksia mm. laatujärjestelmään liittyen.
   - Valvontatoimintaan liittyvien päätöstietojen (esim. laitosmuutostyöt, painelaitetarkastukset, rakennesuunnitelmat) hallinnassa.

2. Uusien YVL-ohjeiden laadinnassa ja voimaansaattamisessa
   - YVL-ohje vaatimusten laadun varmistaminen sekä laitosten auditointi.

3. YVL-ohjeiden sekä kansainvälisten ohjeiden vertailussa
   - olemassa olevien ohjeiden puutteiden tunnistaminen ja täydentäminen nykyistä kattavammaksi.

4. Muussa STUK:n toiminnassa
   - Säteilyturvaohjeiden soveltamisessa/valvonnassa muillakin alueilla (esim. solariumit) kuin ydinvoimalaitosten valvonnassa.

Muussa viranomaistoiminnassa tai valtion laitosten toistuvissa tai suurissa yksittäishankinnoissa (esim. tietojärjestelmät) voidaan vaatimushallintaa myös soveltaa normaalin projektinhallinnan puitteissa.

YVL-ohjeet ovat tavoitteen luonteisia eivätkä varsinaisesti vielä oikeita vaatimuksia. Vaatimukset täsmentyvät vasta niiden tulkinnan yhteydessä. Tällä hetkellä tulkinnat ovat tarkastustoimintaan osallistuvien muistissa eikä niitä ole systemaattisesti kirjattu näkyviin. Vaatimushallinnan puitteissa näitä tulkintoja dokumentoidaan, ideaalitapauksessa jo ennen tarkastusten tekemistä.

Kohdassa 5 kuvattu case-STUK-FIN 5 on käytännön sovellus vaatimushallinnan käytöstä viranomaistoiminnassa.

4.2 Vaatimushallinta voimayhtiöissä

Vaatimushallintaa voidaan soveltaa voimayhtiöissä uusien laitosten sekä laitosuusintoihin liittyvissä hankinnoissa, niihin liittyvissä vaatimusten määrittelyssä tarjouspyyntöä varten sekä tarjousvertailussa ja toimitusvalvonnassa sekä sopimusehtojen määrittelyssä samoin kuin hankintoihin liittyvien alihankintojen hallinnassa.

Lisäksi vaatimushallintaa sovelletaan jo nyt uuden laitoksen tuotantoon valmistautumisessa: käyttötoiminnan ja kunnossapidon käyttöönottovaiheen sekä käytönaikaisen tuen suunnittelussa, siihen liittyvien vaatimusten määrittelyssä tarjouspyyntöä varten, tarjousvertailussa sekä vaatimusten todentamisessa ja kelpuutuksessa.

Viranomaisvaatimusten (lupaprosessit) täyttämiseen liittyvässä valvontatoiminnassa vaatimushallintaa voidaan myös hyödyntää yhdessä lupaviranomaisen kanssa.

5 Vaatimushallinta viranomaisvalvonnassa: case STUK-FIN5

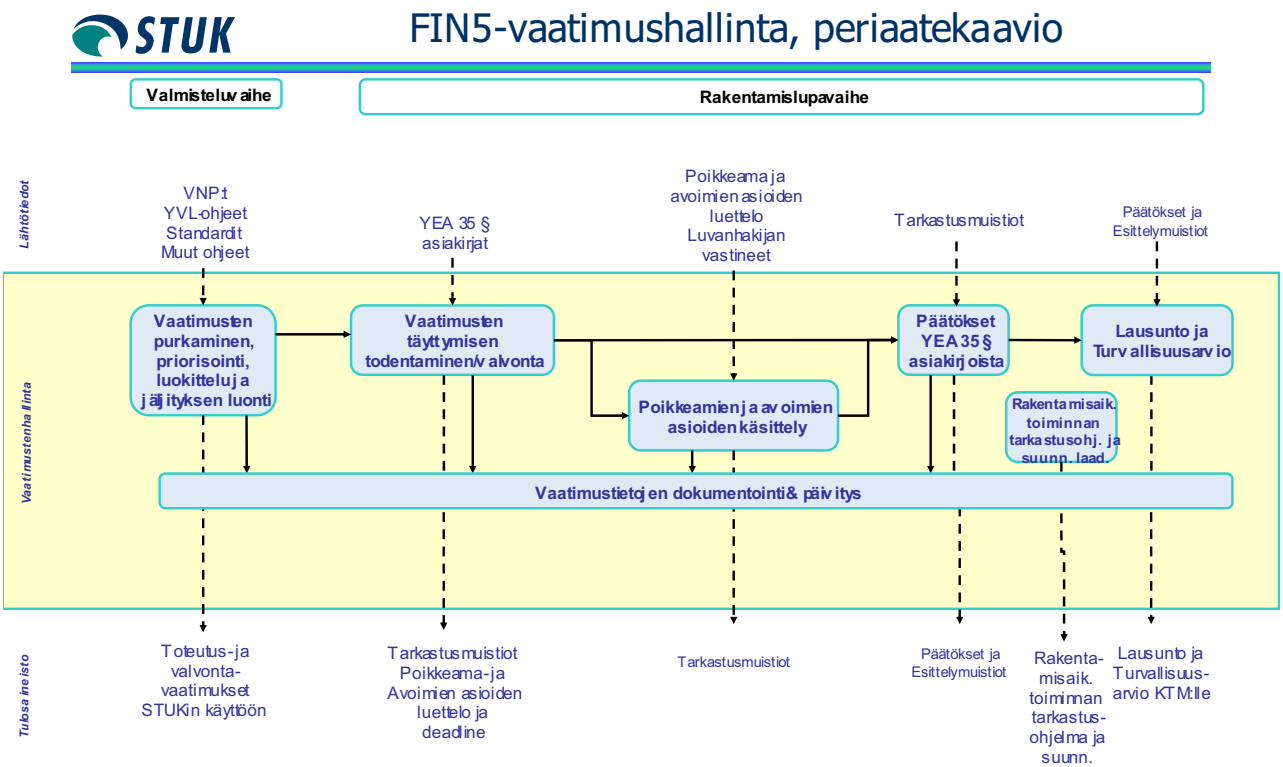
Vaatimushallinnan sovelluskohteeksi oli valittu STUK:n uuteen ydinvoimalaitoshankkeeseen (FIN5) liittyvä ydinturvallisuusvalvonta. Tavoitteeksi asetettiin määritellä alustava vaatimushallintaprosessi. Työ suoritettiin yhdessä STUK:n valvontaprojektiin osallistuvien asiantuntijoiden kanssa.

Työ aloitettiin perehdyttämällä osallistujat vaatimushallintaan ja vaatimusten kirjoittamiseen. Tämän jälkeen tunnistettiin sidosryhmät ja niiden vaatimukset FIN5:n valvonta- ja vaatimushallintatoiminnalle sekä johdettiin niitä ja valvontaprojektin muuta aineistoa ja asiantuntemusta hyväksikäyttäen vaatimushallinnassa tarvittavat toiminnot erikseen valmistelu-, rakentamislupa-, rakentamis- ja käyttölupavaiheille. Liitteessä 5-1 on esitetty valvontaprojektin vaatimushallinnan toimintokuvaus kaaviona.

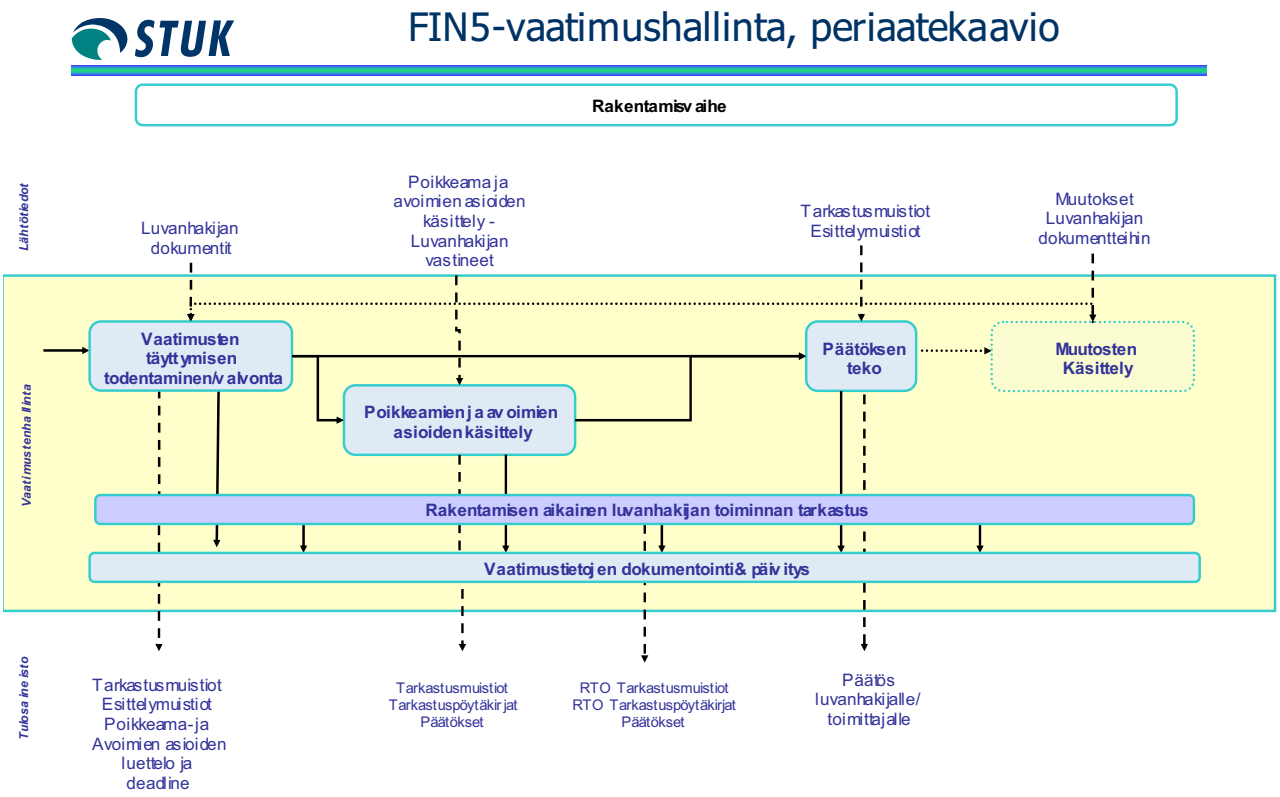
Valmisteluvaiheessa säädöksistä puretaan auki projektin sisäiseen käyttöön toteutus- ja valvontavaatimukset, joiden avulla valvontaa suoritetaan. Kunkin vaatimuksen toteutumista arvioidaan projektin kuluessa sekä kirjataan ylös valvontaan liittyviä tietoja.

Systemaattisen vaatimushallinnan tukemana voidaan valvontaprojektissa seurata koko ajan vaatimusten täyttymistilannetta sekä tarkastustilannetta ja poikkeamia sekä avoimien asioiden tilaa. Säädöksistä puretut vaatimukset selkiyttävät valvontatoimintaa ja tarkastustyöstä jää tiedot, jotka ovat jäljitettävissä aina niiden perustana oleviin ylätason säädöksiin asti. Työn yhteydessä paljastuvat samalla myös säännöstön kehittämistarpeet.

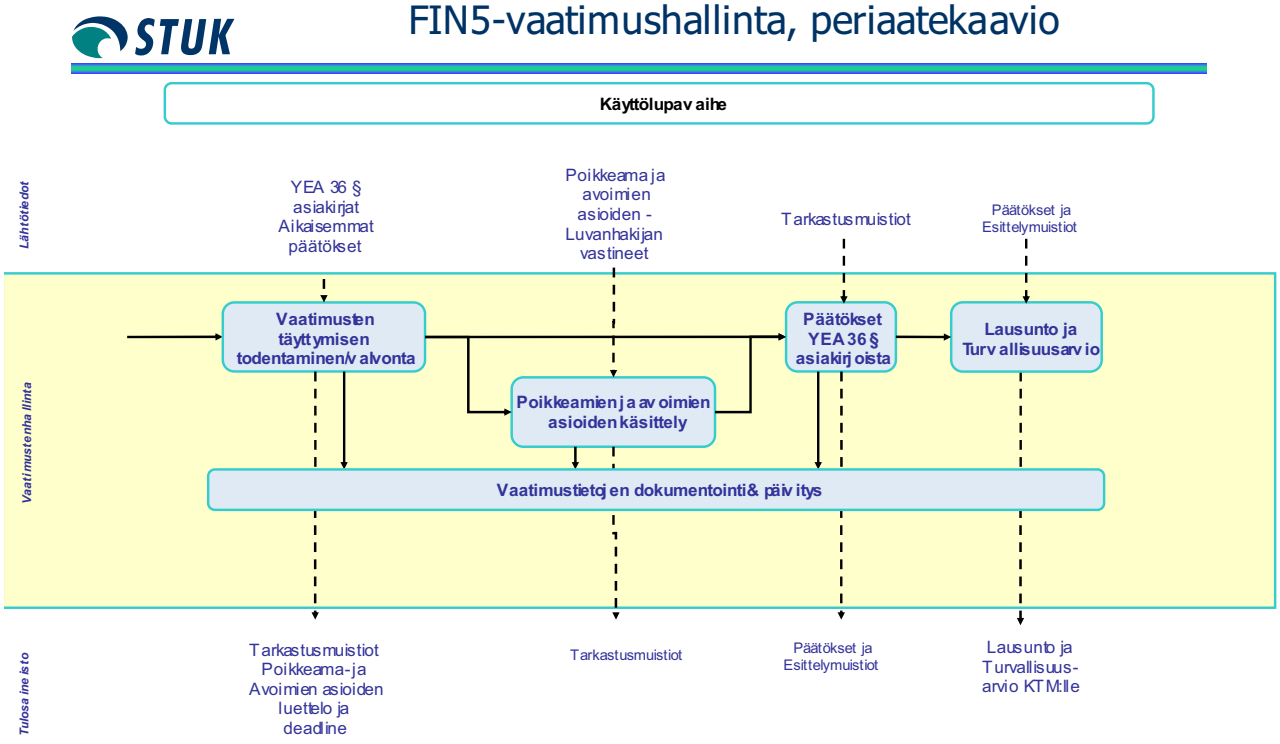
Liite 5-1: **Periaatekaavio 1**



Liite 5-1: Periaatekaavio 2



Liite 5-1: Periaatekaavio 3



6 Vaihe 1, yhteenveto

6.1 Vaatimushallinnan soveltuvuus ydinturvallisuuden parantamisessa

Vaatimushallinnan tutkimuksessa pääpaino on ohjelmistotuotannossa ja siinä nimenomaan ohjelmistojen vaatimusmäärittelyvaiheessa. Muiden järjestelmien kuin ohjelmistojen vaatimushallintaa on tutkittu selvästi vähemmän. Painotusta on ohjannut vaatimusmäärittelyn merkitys järjestelmäsuunnittelussa. Muita vaatimushallinnan osa-alueita – kuten esimerkiksi vaatimusten muutoshallinta, jäljitettävyyden hallinta, vaatimustietojen hallinta – on tutkittu vielä varsin vähän.

Tutkimuksessa on kehitetty yleisiä vaatimusmäärittelyn peruslähestymistapoja sekä vaatimusten kuvaustapoja, jotka soveltuvat käytettäväksi varsin laaja-alaisesti sovellusalueesta riippumatta. Joitakin sovellusaluekohtaisia tutkimuksia on myös tehty.

Vaatimushallinnan käytäntöjä on kehitetty runsaasti keskeisiä peruslähestymistapoja ja niiden hyviä puolia yhdistellen. Pääpaino sovelluksissa on tutkimuksen tapaan ohjelmistotuotannon vaatimushallinnassa, joskin ne ovat yleensä melko yleispäteviä.

Selvityksessä nousi esille tavoitepohjainen lähestymistapa, jota voidaan pitää hyvänä lähtökohtana varmistamaan vaatimusmäärittelyn korkea laatu. Muodolliset vaatimusmäärittelytavat, vaatimusten katselmoinnit sekä vaatimusten jäljitettävyyden hallinta yhdessä tavoitepohjaisen lähestymistavan kanssa ovat hyviä keinoja ohjelmistojen ja ohjelmistopainotteisten järjestelmien, myös turvallisuuskriittisten järjestelmien, luotettavuuden ja laadun varmistamisessa.

Syvällisempi jatkoselvitys on tarpeen kiinnostavimpien käytäntöjen soveltuvuuden selvittämiseksi paremmin tietylle sovellusalueelle.

Ydinturvallisvalvontatoiminnassa sekä voimayhtiöissä löytyy vaatimushallinnan soveltamisalueita mm. YVL-ohjeiden täyttämisen valvonnassa sekä laitosten uus- ja modernisointihankinnoissa.

Case-STUK osoitti, että vaatimushallinnalla on selvästi hyödyllinen rooli osana viranomaisen ydinturvallisuuden valvontatoimintaa. Tällöin varmistetaan valvonnan tarkoituksenmukaisuus ja jäljitettävyys.

6.2 Jatkotoimenpiteet ja suositukset

Kartoituksen yhteydessä varmistui käsitys, että kehittämällä ydinvoimaspesifistä vaatimushallintaa ja sen laajamittaista käyttöönottoa voidaan parantaa Suomen ydinvoimalaitosten turvallisuutta ja taloudellisuutta seuraavilla alueilla:

- ydinturvallisuuden valvontatoiminta
- voimayhtiöiden uudis- ja perusparannushankkeet.

Välitöntä hyötyä on saavutettavissa viranomaisten valvontatoiminnassa. STUK:n YVL-ohjeissa vaatimusmäärittelyä edellytetään käytettäväksi automaatiojärjestelmien ja laitteiden suunnittelussa, toteutuksessa ja käytössä (YVL 5.5). Tämän täyttäminen edellyttää vaatimushallinnan systemaattista soveltamista sekä ydinlaitoksen luvanhakijalta (voimayhtiöt) että valvontaviranomaiselta (STUK).

Lähinnä edellä esitettyä tarvetta palvelemaan ehdotetaan seuraavia jatkoselvityksiä projektin toisessa vaiheessa:

1. Selvitetään perusteellisemmin tavoite- ja skenaario- sekä näkökulmapohjaisia vaatimusmäärittelyn lähestymistapoja ja arvioidaan niiden soveltuvuutta.
2. Selvitetään muodollisia vaatimusmäärittelytapoja sekä agenttipohjaista lähestymistapaa ja arvioidaan niiden soveltuvuutta.
3. Selvitetään vaatimushallintaprosessin muita kuin vaatimusmäärittelykäytäntöjä (lähinnä vaatimusten jäljitettävyyden hallinta, vaatimusten todentaminen ja vaatimusten kelpuutus/katselmointi).

Jatkoselvitykset palvelevat tarvittaessa meneillään olevien uudis- ja perusparannushankkeiden viranomaisvalvontatoimintaa ja voimayhtiöiden investointiprojekteja.

7 Järjestelmävaatimusten määrittelyiden laadinta

7.1 Taustaa

Raportin laatija osallistui Järjestelmävaatimusten analysointi ja vaatimusmäärittelyiden kirjoittaminen -kurssiin (Lontoo, 29.9.–3.10.2003), jonka pitäjänä oli Robert Halligan. Halliganilla on erittäin pitkä ja laaja kokemus suurten yritysten ja laitosten järjestelmäprojekteissa, konsultoinnissa sekä koulutuksessa ja standardointityössä Systems ja Requirements Engineering -alueilla USA:ssa, Australiassa ja Englannissa.

Kurssi keskittyi järjestelmävaatimusten analysointiin ja vaatimusmäärittelyiden (spesifikaatioiden) laadintaan.

7.2 Asiantuntijoiden näkemyksiä vaatimushallinnan soveltamisesta ydinturvallisuusalueella

Erityisen tärkeää asiantuntijoiden mukaan on, että (ydin)turvallisuuteen liittyvien vaatimusten osalta varmistetaan niiden täyttyminen koko järjestelmän hankinta- tai kehitysprosessin ajan. Turvallisuudesta voi olla hyvinkin tarkkoja vaatimuksia järjestelmän suunnitteluun, toteutukseen yms. liittyen, joten niiden täyttyminen on varmistettava huolella. Vaatimusmäärittelyiden laadusta on myös pidettävä erityistä huolta, sillä se vaikuttaa järjestelmän luotettavuuteen ja sitä kautta turvallisuuteen.

Vaatimusmäärittelyyn liittyvät standardit sisältävät yleensä dokumenttien sisältöön liittyviä ohjeita/vaatimuksia (mitä asioita määrittelyissä on oltava), mutta eivät kuvausta siitä prosessista ja niistä menetelmistä, jolla vaatimuksia tuotetaan ja hallitaan koko projektin ajan. Korkealaatuisten vaatimusmäärittelydokumenttien tuottaminen edellyttää myös näiden prosessien ja menetelmien hallintaa.

(Ydin)turvallisuuden vaatimushallinnassa tulisi hyödyntää erityisesti seuraavia myöhemmin lyhyesti kuvattavia järjestelmävaatimusmäärittelyn elementtejä:

- Toiminnallinen analyysi
- Skenaarioanalyysi
- Viranomaisvaatimukset
- Out–of–range-analyysi

joiden avulla saadaan erityisesti turvallisuuteen liittyviä vaatimuksia tunnistettua ja määriteltyä.

7.3 Kurssilla esitetty vaatimusmäärittelykonsepti

7.3.1 Vaatimusmäärittelyiden laadun merkitys

Vaatimusmäärittelydokumenteissa näkyy selkeästi se, että ihmiset ajattelevat yleensä fysikaalisesti ja ratkaisuorientoituneesti, jonka seurauksena dokumenteissa on esitetty runsaasti suunnitteluratkaisuja sekä usein tarpeettomia, ratkaisuvaihtoehtoja rajaavia reunaehtoja pelkkien vaatimusten sijasta.

Vaatimusten määrittely edellyttää abstraktia ajattelukykyä, mikä on laatijoille hyvin usein vaikeaa. Dokumenteista näkyy myös se, että kaikkia olemassa olevia, ”piileviä” vaatimuksia ei ole tunnistettu eikä dokumentoitu. Nämä vaikuttavat sekä hankittavan järjestelmän ominaisuuksiin että hankinnassa mukana olevien osapuolten työn vaikeuteen: huonolaatuisia vaatimuksia on työlästä ja hankalaa tulkita sekä täyttää. Tämä aiheuttaa osaltaan lisätyötä, kun on haettava lisätietoja tarjouksen pyytäjältä. Myös vaatimusten todentaminen on vaikeaa huonolaatuisten vaatimusmäärittelyjen perusteella.

Järjestelmien tarjouspyynnöissä vaatimusmäärittelyt muodostavat tärkeän osan ja niiden laatu määrittää paljolti hankintaprojektin sujumisen ja onnistumisen sekä hankittavan järjestelmän toimivuuden ja laadun.

Seuraavassa esitettävä, kurssiin pohjautuva

malli ja siihen liittyvät käytännöt muodostavat hyvän pohjan korkealaatuisten järjestelmävaatimusten määrittelyyn ja analysointiin sekä vaatimusmäärittelydokumenttien laatimiseen.

7.3.2 Järjestelmävaatimusanalyysi

Järjestelmävaatimusten käyttäjiä ovat järjestelmän käyttäjät (järjestelmän on vastattava heidän tarpeitaan) ja hankkijat (vaatimukset ovat sopimus siitä, mitä on toimitettava), hankkijan sopimusvastaavat (vaatimukset ovat sopimusehtojen perusta), toimittaja (vaatimukset muodostavat toimitussopimuksen ehdot, asiakastyytyväisyyden mittarit, toimituksen optimointiperustan) sekä hankkijan ja toimittajan laadunvarmistushenkilöstö (vaatimukset muodostavat toimittajan laatu-/suorituskykyvaatimuskriteerit).

Vaatimusten laatuun ja hallintaan liittyvät puutteet on osoitettu monissa tutkimuksissa projektien epäonnistumisten suurimmaksi syyryhmäksi. Siten panostaminen vaatimusmäärittelyihin ja vaatimushallintaan johtaa monessa suhteessa parempaan lopputulokseen (laatu, aikataulu, kustannus).

Järjestelmävaatimusanalyysin tavoitteena on saada aikaan joko uudet vaatimusmäärittelyt tai saada parannettua mahdollisesti jo olemassa olevia vaatimusmäärittelyjä siten, että ne vastaavat vaatimuksille asetettuja laatuvaatimuksia (yksikäsitteisyys, selkeys, kattavuus, yms.), mikä muodostaa hankinnalle hyvän perustan.

Järjestelmävaatimusanalyysillä voidaan

- osoittaa projektin riskialttius ja pienentää projektin riskejä
- tunnistaa puutteet vaatimusmäärittelyissä sekä parantaa vaatimusmäärittelydokumenttien laatua
- parantaa projektin/hankkeen onnistumisen todennäköisyyttä
- vaikuttaa hankittavan/kehitettävän järjestelmän luotettavuuteen ja muihin laatuominaisuuksiin.

Järjestelmävaatimusanalyysi voi olla laajimmillaan monivaiheinen ja vaatia paljon työtä, jos lähtökohtana ovat huonosti laaditut määrittelyt.

Järjestelmävaatimusten analysoinnissa ja määrittelyissä voidaan käyttää seuraavia, tilanteen mukaan valikoiden sovellettavia menettelyjä ja käytäntöjä (kuva 2).

1. **Käyttö- ja tukikonseptin** määrittely tai parantaminen
   - konsepti kuvaa hankittavan järjestelmän tai palvelun suunniteltua käyttö- ja ylläpito- tai tukitapaa ja auttaa ymmärtämään suunniteltavan järjestelmän/palvelun peruslähtökohdat, -oletukset ja rajaukset
   - mikäli konseptia ei ole laadittu jo järjestelmähankkeen alussa ennen sidosryhmävaatimusten määrittelyä, on se syytä tehdä viimeistään järjestelmävaatimusmäärittelyvaiheessa.

2. **Sidosryhmäanalyysi**
   - analyysin tunnistetaan hankittavan järjestelmän tai palvelun sidosryhmät ja heidän intressinsä sekä priorisoidaan sidosryhmät
   - analyysin paras ajoitus olisi jo ennen sidosryhmävaatimusten määrittelyä.

3. **Vaatimusten laatuanalysointi**
   - analyysi auttaa jo määriteltyjen vaatimusten laadun arvioinnissa ja niiden puutteiden sekä vaatimusmäärittelydokumentin analysointitarpeen tunnistamisessa
   - analyysi auttaa myös täydentämään ja parantamaan vaatimusilmaisuja
   - käytetään silloin, kun on todettu, että laadittu vaatimusmäärittelydokumentti on laadultaan riittämätön ja se voi muodostaa suuren riskin koko projektin onnistumiselle.

4. **Kontekstianalyysi**
   - analyysillä tunnistetaan järjestelmän liitynnät ympäröivään maailmaan, mikä auttaa myöhemmässä vaiheessa tunnistamaan näihin liityntöihin liittyviä vaatimuksia.

5. **Suunnitteluvaatimusten analyysi**
   - analyysin tarkoituksena on tarpeettomien suunnittelurajoitusten ja -ristiriitojen poisto muuntamalla rajoitukset toiminnallisiksi vaatimuksiksi sekä täydentämällä toiminnallisia vaatimuksia.

6. **Tila- ja toimintatila-analyysi**
   - analyysillä tunnistetaan suunniteltavan järjestelmän tilat ja toimintatilat (toimintamoodit)

- analyysi täydentää järjestelmän toiminnan kuvausta sekä auttaa tunnistamaan niiden kautta lisää puuttuvia vaatimuksia.

7. **Toiminnallinen analyysi**
   - analyysi kuvaa kaikki järjestelmän elinkaaren aikaiset toiminnot ja auttaa täydentämään toiminnallisia vaatimuksia.

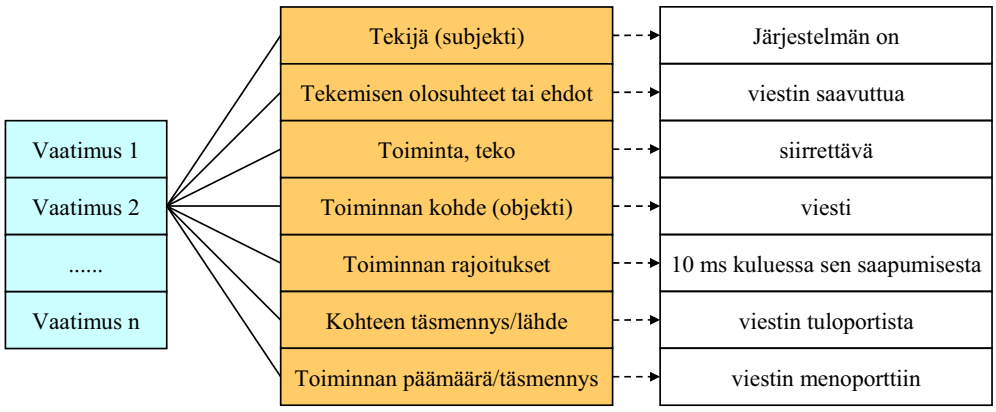
8. **Vaatimusten jäsennysanalyysi**
   - analyysissä jäsennetään olemassa olevia vaatimuksia ja saatetaan ne kieliopillisesti ja merkitykseltään oikeaan muotoon
   - kuvassa 1 on esitetty vaatimusilmaisun rakennemalli, jota käytetään jäsennysanalyysissä.

9. **Kohdeanalyysi**
   - analyysi tuottaa kohdemallin, jossa on kuvattu järjestelmän elementit (yleensä data) ja niiden väliset loogiset ja fyysiset suhteet sekä attribuutit (käytetään yleisesti IT-järjestelmille)
   - tuottaa tietoa ulkoisiin liityntöihin liittyvien sekä joidenkin toiminnallisten vaatimusten tunnistamiseksi.

10. **Muiden reunaehtojen tunnistaminen**
    - analyysillä tunnistetaan vielä puuttuvia reunaehtoja
    - auttaa täydentämään ei-toiminnallisia vaatimuksia.

11. **Skenaarioanalyysi**
    - skenaario kuvaa järjestelmän ja/tai sen toimijoiden (esim. ihmiset, ohjelmisto, laite) peräkkäisiä ja rinnakkaisia aktiviteetteja tai toimintoja
    - analyysi auttaa määrittelemään järjestelmän eri toimintaskenaarioihin liittyviä vaatimuksia
    - skenaariot voivat sisältää mm. järjestelmän suunnitellut toiminnat, erilaiset ympäristöolosuhteet, vahingot ja kaikki muut poikkeamat tai vaihtelut järjestelmän käytössä, ylläpidossa ja käytöstä poistossa
    - 1. tason skenaarioanalyysi on osa toiminnallista analyysiä
    - 2. tason skenaarioanalyysissä tunnistetaan 1. tason toiminnalliseen malliin liittyvien erilaisia ympäristöolosuhteita sekä niihin liittyviä ympäristö- ja liityntävaatimuksia.

12. **Out–of–range-analyysi**
    - analyysi tunnistaa järjestelmän käytön/tuen/käytöstä poiston epänormaaleihin (sallittujen rajojen ulkopuolisiin) arvoihin liittyviä vaatimuksia
    - analyysi suoritetaan vaatimusanalyysin viime vaiheissa käymällä läpi kaikki ”normaalit” vaatimukset ja tunnistamalla eisallitut ulkoiset olosuhteet tai liityntäparametrit, joihin liittyvät vaatimukset tulevat näin esille (on eräänlaista riskianalyysiä).



Kuva 1. Vaatimuksen rakennemalli jäsennysanalyysissä.

13. **Sidosryhmäarvoanalyysi**
    - analyysillä määritetään järjestelmän ominaisuuksien arvo sidosryhmille
    - vaikuttaa järjestelmävaatimuksiin.

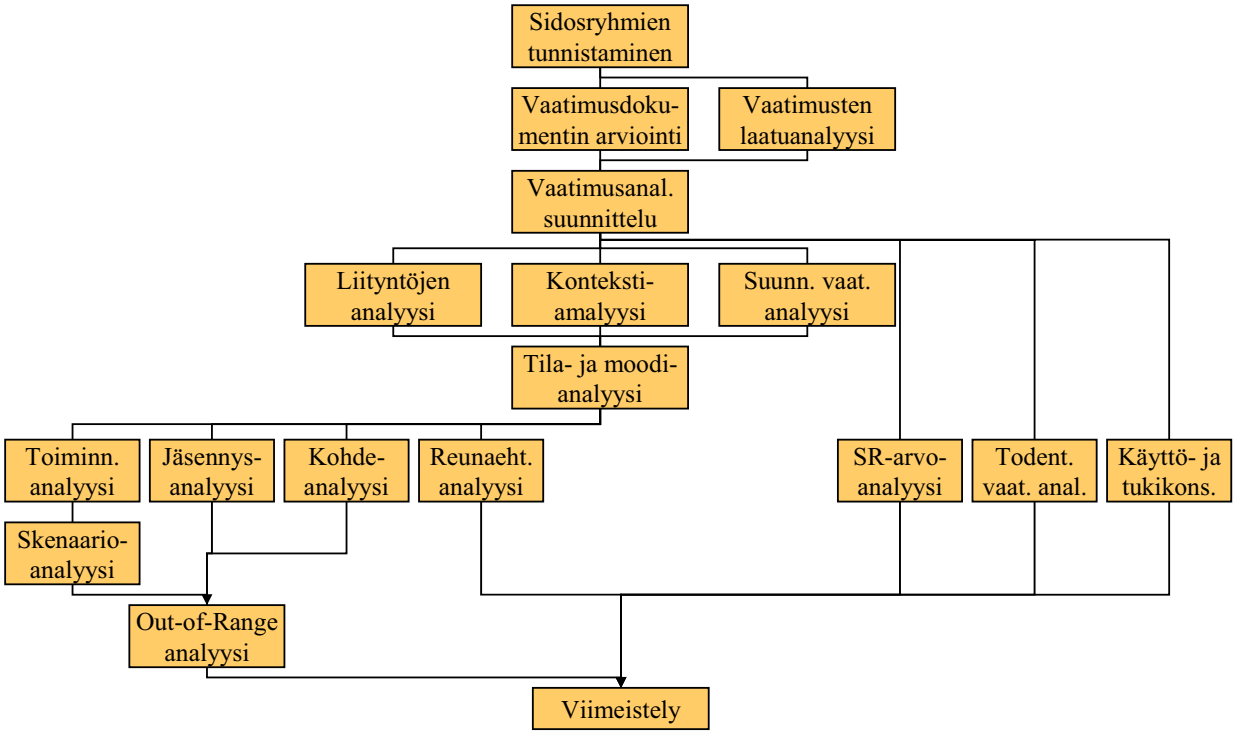
14. **Todentamisvaatimusten analyysi**
    - analyysillä tunnistetaan järjestelmävaatimusten todentamiseen liittyvät todentamistavat ja määritetään todentamisvaatimukset.

15. **Vaatimusmäärittelyjen todentaminen ja kelpuutus**
    - todentamistapojen tunnistamisella jo vaatimusmäärittelyvaiheessa voidaan varmistaa vaatimuskuvausten laatu ja auttaa tunnistamaan puuttuvia vaatimuksia sekä poistamaan päällekkäisiä vaatimuksia
    - järjestelmävaatimusten todentaminen suoritetaan sidosryhmävaatimuksia vastaan
    - kelpuutuksella osoitetaan vaatimusten olevan riittävän laadukkaita jatkotyöskentelyn pohjaksi (suunnitteluvaihe).

Vaatimusten analysoinnissa käytetään jo usein tekstipohjaisia, joskus jopa tekoälyyn perustuvia vaatimusanalysointityökaluja, joissa voi olla kielen rakenteeseen ja kielioppiin perustuvia menetelmiä puutteellisten vaatimusilmaisujen automaattiseksi tunnistamiseksi (esim. SIR/REX). Tällaiset työkalut nopeuttavat rutiinitapausten hallintaa.

Vaatimusanalyysiprosessia sovelletaan suunnitteluvaiheessa monilla eri tasoilla, kuten järjestelmä-, osajärjestelmä-, laite- ja komponenttitasoilla. Prosessi on koko suunnitteluprosessin aikaista jatkuvaa toimintaa, joka limittyy läheisesti suunnittelutoimintaan ja on luonteeltaan iteratiivista. Projektin edetessä vaatimusmäärittelyn osuus vähenee ja suunnittelun osuus lisääntyy.

Em. menettelyjen avulla saadaan aikaan kattavat ja korkealaatuiset järjestelmävaatimusmäärittelyt. Ne toimivat pohjana tarjouspyyntöjen vaatimusmäärittelyosioille ja järjestelmän hankinta-/toimitussopimukselle sekä järjestelmän toimitusvalvonnalle, todentamiselle ja kelpuutukselle.



Kuva 2. Järjestelmävaatimusten määrittämismenettelyt.

7.3.3 Vaatimusmäärittelyiden (-spesifikaatioiden) kirjoittaminen

Järjestelmän vaatimusmäärittelydokumenttiin kirjataan, kootaan ja organisoidaan järjestelmävaatimusmäärittelyillä tuotetut vaatimustiedot helposti luettavaan ja kaikkien osapuolten ymmärtämään muotoon. Dokumentoinnissa käytetään tarvittaessa apuna standardeja. Standardit muodostavat hyvän tuen dokumenttien sisällön määrittelylle, mutta ne eivät ole yleensä ole täysin kattavia. Raportin osassa 8 esitetyssä IEEE Std 830-1998 standardissa on esitetty eräs vaatimusmäärittelydokumentin rakennemalli.

Hyvin suoritettu vaatimusmäärittely tai -analyysi tuottaa tarvittaessa standardien tukemana hyvän lopputuloksen sekä hankkijan, tarjoajien että toimittajien kannalta. Hyvä vaatimusmäärittelydokumentti kuvaa hankittavan järjestelmän todelliset vaatimukset kattavasti ja auttaa aikaansaamaan halutunlaisen lopputuloksen sekä helpottaa kunkin osapuolen työtä.

Vaatimusspesifikaatioiden rakenteella on merkitystä ymmärrettävyyden kannalta, missä havainnollisuus, loogisuus, kielellinen oikeellisuus ja tarkkuus, termistön selkeys sekä sisällön laatu ovat tärkeitä tekijöitä, joita ei saa unohtaa vaatimusmäärittelydokumenttia laadittaessa ja arvioitaessa.

8 Standardit ydinvoimalaitosten turvallisuuteen liittyvissä järjestelmissä

8.1 Turvallisuuteen liittyvät standardit

8.1.1 Yleistä

Vaatimushallintaan ja ydinvoiman turvallisuuteen liittyviä standardeja tunnistettiin yhdessä STUK:n asiantuntijoiden kanssa. Tässä kuvataan pääosin standardien arviointia vaatimushallinnan näkökulmasta STUKin standardiaineistosta kerättyyn tietoon perustuen.

Standardeja on tarkasteltu vaatimushallintanäkökulmasta ja taustalla on kattava vaatimushallintaprosessi, johon kuuluvat seuraavat osaprosessit:

- sidosryhmävaatimusten määrittäminen
- järjestelmävaatimusten määrittäminen
- järjestelmän/osien todentaminen
- järjestelmän/osien kelpuutus
- vaatimusten jäljitettävyyden hallinta
- vaatimusten muutosten hallinta.

Sidosryhmävaatimusten määrittäminen puuttuu useimmista standardeista. Se on kuitenkin tarpeellinen erityisesti suurten ja monimutkaisten järjestelmien hankinnassa ja toteuttamisessa, koska sidosryhmätarpeiden ja -vaatimusten vajavainen tunteminen ja täyttäminen voi aiheuttaa järjestelmän elinkaaren myöhemmissä vaiheissa monia ongelmia, joista osa liittyy järjestelmien turvallisuuteen (esim. valvomosuunnittelussa oleellisia ovat käyttäjävaatimukset).

Käsiteltyjen standardien luettelo on esitetty standardiviiteluettelossa raportin osan 8 lopussa. Liitteessä 8-1 on esitetty yhteenvetona raportin laatijan kvalitatiivinen arviointi, miten kattavasti kukin standardi sisältää vaatimushallintaa.

8.1.2 Standardit vaatimushallintanäkökulmasta

Liitteessä 8-1 on esitetty analysoitujen standardien ”hyvyys” vaatimushallintanäkökulmasta arvioituna asteikolla 1–3 (tyydyttävä – hyvä – erittäin hyvä). Standardit kattavat eri laajuudelta vaatimushallintaprosessia. Liitteen 8-1 taulukossa ne standardit, jotka eivät kata koko prosessia, on arvioitu myös ao. osakokonaisuuden kannalta. Seuraavassa esitellään lyhyesti analysoidut standardit tai niihin verrattavat säädökset tai ohjeet.

CMMI-standardit

CMMI[8-1] on Software Engineering Instituten kehittämä prosessien kypsyyden arviointiin ja parantamiseen tarkoitettu standardinomainen malli (Capability Maturity Model, CMMI). Sen perusajatuksena on toimintaprosessien kehittämisen kautta varmistaa organisaation tuottamien tulosten (esim. ohjelmisto) laatu. CMMI soveltuu kaikkien järjestelmien (ml. ohjelmistot) kehittämisprosessien parantamiseen.

Näkökulmana voi olla kaikkien organisaation prosessien parantaminen, jolloin on kyseessä yrityksen kypsyystason määrittäminen 5-portaiselta kypsyystasoasteikolta. Toinen näkökulma on yksittäisten toimintaprosessien (esim. vaatimushallinta) kyvykkyystason määrittäminen sekä parantaminen. CMMI kuvaa hyvien käytäntöjen mallin, jota vasten voidaan verrata olemassa olevia käytäntöjä sekä määrittää toiminnan tai toimintaprosessien kehittämistavoitteet. Se sisältää myös käytännöt kehitettyjen prosessien käyttöönottoon.

CMMI-malli on integroitu malli, jossa on yhdistetty aikaisemmat SW (Software), SE (Systems Engineering) ja IPPD (Integrated Product Development) -mallit.

CMMI kattaa kaikki vaatimushallinnan osavaiheet (ml. myös sidosryhmävaatimusten määrittäminen, jäljitettävyyden ja vaatimusten muutosten hallinta, elinkaari) ja on analysoiduista standardeista tässä suhteessa kaikkein kattavin.

EUR-vaatimukset

EUR Requirements[8-2] on ydinvoimalaitosten I&C -järjestelmien hankintaan ja suunnitteluun sekä toimituksiin (osana laitoshankintaa) tarkoitettu standardinomainen vaatimusmäärittelydokumentti. Se sisältää I&C- järjestelmien ja Man-Machine-liityntöjen suunnitteluperiaatteita, I&C-järjestelmän toiminnalliset vaatimukset sekä toteutusvaatimukset ja vaatimuksia toiminnalliselle analyysille ja toimintojen allokoinnille. Se sisältää hyvin lyhyesti myös vaatimusmäärittelyn ja V&V-vaatimukset (V&V: todentaminen ja kelpuutus) osana suunnittelu- ja toteutusprojektia. Dokumentti ei edellytä järjestelmän toteutuksessa noudatettavaksi erityistä vaatimushallintaa.

IAEA-turvallisuusohjeet (Safety Guides)

IAEA NS-R-1 [8-3] määrittelee ydinvoimalaitosten ydinturvallisuuden suunnitteluvaatimukset, joita on sovellettava turvallisuustoiminnoille sekä niitä vastaaville rakenteille, järjestelmille ja komponenteille sekä proseduureille. Pääsisältökohtia ohjeessa ovat turvallisuustavoitteet ja -konseptit, turvallisuusjohtamisen vaatimukset, tekniset perusvaatimukset sekä laitoksen ja järjestelmien suunnittelun vaatimukset. Tässä ohjeessa ei ole kuvattu suunnittelu- eikä vaatimushallintaprosessia, mitkä on sisällytetty ohjeisiin [8-4] ja [8-5].

IAEA NS-G-1.1 [8-4] opastaa ydinvoimalaitosten turvallisuuteen liittyvien tietokonejärjestelmien turvallisuuden osoittamiseen liittyvien todisteiden keräämisessä sekä siihen liittyvän dokumentaation tuottamisessa. Ohje täydentää IAEA NS-R-1:stä. Pääpaino on turvallisuuteen liittyvien tietokonepohjaisten järjestelmien turvallisuuden ja luotettavuuden osoittamiseen liittyvän dokumentaation tuottamisessa.

Ohje sisältää sekä ohjelmiston että sen liittämisen vastaavaan tietokonejärjestelmään. Ohjelmiston osalta se kattaa sen tuottamiseen, arviointiin ja lisensiointiin (uudet ja olemassa olevat ohjelmistot) liittyvät vaatimukset. Ohje sisältää järjestelmän koko suunnitteluprosessin – sisältäen vaatimusmäärittelyn, todentamisen ja kelpuutuksen – kuvauksen sekä vaatimukset tietokonepohjaisten järjestelmien turvallisuuden johtamiselle ja laadunvarmistukselle sekä dokumentoinnille. Lisäksi se kattaa myös järjestelmän integrointiin, asennukseen, käyttöön ja modifikaatioihin liittyviä vaatimuksia.

Ohje sisältää yleisluonteisesti vaatimushallintaprosessin tärkeimmät vaiheet (järjestelmävaatimusten määrittäminen, todentaminen, kelpuutus sekä jäljitettävyyden hallinta), mutta ei sidosryhmävaatimusten määrittelyä eikä vaatimusten muutoshallintaa.

IAEA NS-G-1.3 [8-5] kuvaa, miten ydinvoimalaitosten turvallisuudelle tärkeiden I&C-järjestelmien vaatimukset pitäisi täyttää ja suunnitella. Ohje sisältää vaatimuksia I&C-järjestelmien suunnitteluprosessille sekä yleisiä ja erityisiä suunnitteluperiaatteita turvajärjestelmille sekä turvallisuuteen liittyville järjestelmille. Ohjeessa on myös suunnitteluprosessin lyhyt kuvaus sekä vaatimuksia dokumentoinnille. Ohje täydentää ohjetta IAEA NS-R-1.

Suunnitteluprosessiin sisältyvä vaatimushallintaprosessi on kuvattu hyvin lyhyesti ja yleisluontoisesti eikä se sisällä vaatimusten muutoshallintaa eikä vaatimusten jäljitettävyyden hallintaa.

IEC–standardit

IEC on laatinut useita standardeja koskien järjestelmien turvallisuutta. Näistä osa koskee ydinvoimalaitosten turvallisuuteen liittyviä järjestelmiä osan ollessa yleisluonteisia ja sopien siten kaikille sovellusalueille.

IEC 60880 [8-6] on tarkoitettu korkeaa luotettavuutta edellyttäville ohjelmistoille, joita käytetään ydinvoimalaitosten turvallisuusluokan A toiminnoissa (IEC luokittelun mukaan).

Standardi esittää vaatimukset ohjelmiston kehittämisen, käytön ja ylläpidon vaiheille lähtien liikkeelle turvallisuusjärjestelmän (yläjärjestelmä) vaatimuksista. Se sisältää yleisellä tasolla myös vaatimushallintaprosessin elementit tehtävämuodossa lukuun ottamatta sidosryhmävaatimusten määrittelyä sekä jäljitettävyyden ja systemaattista vaatimusten muutosten hallintaa.

Standardi sisältää myös yleisiä suunnitteluperiaatteita mm. man–machine-liitynnälle, mitkä pitävät sisällään eräällä tavalla käyttäjävaatimuksia yleisessä muodossa.

IEC 60880-2 [8-7] täydentää IEC 60880-standardia. Se käsittää turvallisuusluokkaan A (IEC) kuuluvien ohjelmistojen aiheuttamien yhteisvikojen välttämisperiaatteita, ohjelmistojen kehittämisen automatisointityökaluihin ja niiden kel-

puutukseen ja testaukseen liittyviä vaatimuksia sekä olemassa olevien ohjelmistojen kelpuutukseen liittyen ohjelmien ja niiden käyttökokemusten arviointi- sekä integrointivaatimukset.

Standardi ei tarkastele asioita vaatimushallintanäkökulmasta, mutta taustalla on kuitenkin IEC 60880:n sisältämä vaatimushallinta.

CEI/IEC (60)964 [8-8] määrittelee vaatimukset ydinvoimalaitoksen valvomon man–machine-liitynnälle. Standardi sisältää valvomon suunnitteluperusteita, toiminnallisen analyysiin ja toiminnallisten suunnitteluspesifikaatioiden laadintaan sekä valvomojärjestelmän todentamiseen ja kelpuutukseen liittyviä vaatimuksia. Standardia ei ole tehty vaatimushallintanäkökulmasta, mutta toiminnot ja niiden suorituskykyarvot vastaavat jossain määrin toiminnallisia vaatimuksia. Todentaminen ja kelpuutus on kuitenkin esitetty.

IEC 60780 [8-9] on tarkoitettu ydinvoimalaitosten turvallisuusjärjestelmien sähkölaitteiden (hardware)kelpuutukseen. Se sisältää yleisen kelpuutusprosessin, -proseduurit ja menetelmät. Kelpuutus on osa vaatimushallintaa.

IEC 60987 [8-10] sisältää ydinvoimalaitoksen turvallisuudelle tärkeiden järjestelmien tietokonelaitteiden (hardware) elinkaaren eri vaiheisiin liittyviä sekä dokumentointiin liittyviä vaatimuksia. Vaatimushallinta ei standardissa ole riittävässä määrin esitetty ja siitä puuttuu vaatimusten jäljitettävyyden ja muutosten hallinta.

IEC 61226 Ed. 1.0 [8-11] esittää luokittelumenetelmän ydinvoimalaitoksen I&C-toiminnoille ja laitteille, jotka kuvaavat turvallisuustoiminnon tärkeyttä. Luokitus määrää sitten sovellettavat suunnittelukriteerit. Standardi sisältää myös suunnitteluvaatimukset (yleiset ja erityiset) eri luokkien järjestelmille ja laitteille. Ei sisällä vaatimushallintaa.

IEC 61508 [8-12] esittää vaatimuksia prosessiteollisuuden turvallisuustoimintoja hoitaville sähköisille, elektronisille ja ohjelmoitaville S/E/OE-järjestelmille (sähkö/elektroniikka/ ohjelmoitava elektroniikka). Standardin 1. osa määrittelee vaatimukset turvallisuusvaatimusten määrittämiseksi järjestelmälle ja niiden allokoimiseksi S/E/OE-järjestelmille, muille järjestelmille sekä muille riskienhallintaelementeille. 2. ja 3. osassa on esitetty laitteiden ja ohjelmistojen kehittämisvaatimukset. Standardi koostuu seitsemästä osasta.

Standardin 2. ja 3. osa sisältävät yleiset vaatimushallintaprosessin mukaiset tehtävät (laitteiston/ohjelmiston turvallisuus, vaatimusmäärittely ja allokointi, turvallisuuden kelpuutus, turvallisuuden todentaminen, integroinnin todentaminen, turvallisuuden kelpuutus), mutta siitä puuttuvat sidosryhmävaatimusten määrittäminen, jäljitettävyyden ja muutosten hallinta.

IEC 61513 [8-13] sisältää vaatimuksia I&C-järjestelmien (sekä analogisten [hard-wired] että tietokonepohjaisten tai niiden kombinaatioiden) arkkitehtuurin suunnitteluun ja I&C-toimintojen allokointiin I&C-järjestelmille sekä vaatimuksia niiden muille elinkaaren vaiheille. Standardi esittelee konseptin I&C-järjestelmän arkkitehtuurin turvallisuuden sekä yksittäisten järjestelmien elinjaksolle.

Standardi soveltuu sekä uusille että käyville (uudistukset) laitoksille. Se korostaa täydellisiä ja tarkkoja vaatimuksia, jotka on johdettu laitoksen turvallisuustavoitteista, edellytyksenä kattavien vaatimusten kehittämiselle I&C-järjestelmän arkkitehtuurille ja samalla turvallisuuteen liittyvien I&C-järjestelmien kehittämiseksi. I&C-järjestelmävaatimukset johdetaan laitoksen turvallisuussuunnitteluperusteista ja dokumentoidaan. Standardi sisältää melko yleisellä tasolla vaatimushallintaa (järjestelmävaatimusmäärittelyn, todentamisen ja kelpuutuksen turvallisuusluokan A ja B järjestelmille (IEC).

IEC 61839 [8-14] määrittelee proseduurit toiminnallisen analyysille ja toimintojen kohdentamiselle automaatiojärjestelmille ja ihmisille ydinvoimalaitoksen valvomon suunnittelussa, joita vaaditaan IEC 60964:ssa. Standardi ei sisällä erityistä vaatimushallintanäkökulmaa, mutta toiminnallinen analyysi on kuitenkin yksi vaatimusmäärittelyn apuväline.

IEC 62138 [8-15] esittää tietokonepohjaisten I&C-järjestelmien ohjelmistojen – jotka tukevat B- ja C-kategorian (IEC) turvallisuustoimintoja – vaatimukset. Se kattaa toimintojen yleiset vaatimukset, valmisohjelmistojen valintavaatimukset, ohjelmiston vaatimusmäärittelyn vaatimukset, ohjelmiston suunnittelun vaatimukset, uuden ohjelman toteutusvaatimukset, ohjelmistonäkökohdat järjestelmäintegraatiossa ja ohjelmistomuutokset. Standardi täydentää IEC 60880 ja IEC 60880-2 standardeja ja on yhdenmukainen IEC 61513:n kanssa. Vaatimushallinnan perusprosessit on esitetty lyhyesti lukuun ottamatta sidos-

ryhmävaatimusten määrittelyä ja vaatimusten jäljitettävyyden sekä muutosten hallintaa.

IEC 9126 [8-16] on tarkoitettu ohjelmiston laatuvaatimusten määrittelyyn ja ohjelmiston arviointiin. Se täydentää vaatimusmäärittelyvaihetta kuvaamalla laatuvaatimusten määrittelytavan, mikä on vain pieni osa vaatimushallintaa.

ISO/IEC 12207 [8-17] sisältää ohjelmiston hankinta-, toimitus-, kehitys-, käyttö- ja ylläpitoprosessien sekä tukiprosessien kuvauksen. Se on hyvä ohjelmistojen elinjaksoprosessistandardi. Se sisältää yleiset vaatimushallintatehtävät, mutta ei sidosryhmävaatimusten määrittelyä eikä vaatimusten jäljitettävyyden ja muutosten hallintaa.

ISO/IEC 15288 [8-18] kuvaa järjestelmän elinjaksoprosessit kattaen järjestelmän hankinta-, toimitus-, projektinhallinta-, järjestelmän kehitys-, käyttö- ja ylläpitoprosessien sekä organisaation prosessien kuvauksen. Siinä on esitetty myös järjestelmien elinkaaren vaiheet sekä järjestelmämallit liitteenä sekä myös liitynnät ISO/IEC 12207:een (ohjelmistojen vastaava standardi). Se sisältää myös vaatimushallintaprosessin seuraavat osat sidosryhmävaatimusten ja järjestelmävaatimusten määrittely (vaatimusanalyysi), todentaminen, kelpuutus ml. jäljitettävyyden hallinnan. Ainoastaan muutosten hallinta puuttuu. Prosessit on esitetty kuitenkin melko yleisellä tasolla.

IEEE-standardit

ANSI/IEEE 1008 [8-19] esittää ohjelmiston yksikkötestausten suunnitteluun, toteutukseen ja arviointiin liittyvät vaatimukset. Testaukset ovat osa vaatimushallintaa.

ANSI/IEEE Std 829-1983 [8-20] on tarkoitettu ohjelmistojen testausten suunnitteluun, määrittelyyn ja testausdokumenttien laadintaan.

IEEE 1012 [8-21] sisältää ohjelmiston todentamis- ja kelpuutusprosessin (V&V) koko ohjelmiston elinkaaren aikana. Se sisältää ohjelmiston hankintaan, toimitukseen, kehittämiseen, käyttöön ja ylläpitoon liittyvät V&V-tehtävät lähtö- ja tulostietoineen. Vaatimushallintamielessä tämä sisältää analysoitujen standardien kattavimman V&V-prosessin sisältäen myös jäljitettävyyden hallinnan. V&V-tehtävät määräytyvät ohjelmiston eheystasojen (eheystasot ovat kriittisyysluokkia) mukaan. Se on esim. yhdessä IEC 12207 kanssa hyvä yhdistelmä ohjelmistotuotannossa.

IEEE 1074 [8-22] on tarkoitettu ohjelmistotuotannon prosessien parantamiseen ohjelmistoprojekteja johtaviin ja toteuttaviin organisaatioihin (prosessiarkkitehdille), mutta myös prosessin toteuttajille. Standardi sisältää tarvittavaa informaatiota elinjaksoprosessien määrittämiseen sekä vaatimukset ohjelmiston elinjaksoprosessien määrittämiselle. Yleinen vaatimushallintaprosessi sisältyy aktiviteetteihin lukuun ottamatta sidosryhmävaatimusten määrittelyä eikä vaatimusten jäljitettävyyden ja muutosten hallintaa.

IEEE 7.4.3.2 [8-23] esittää ydinvoimalaitosten turvajärjestelmien tietokoneiden toiminnalliset ja suunnitteluvaatimukset. Standardi täydentää IEEE Std 603-1998:n vaatimuksia mm. ohjelmiston laatuun ja järjestelmän eheyteen (integrity) liittyviltä osin. Vaatimushallintaan liittyviä asioita standardissa on esitetty melko suppeasti.

IEEE Std 1028-1997 [8-24] esittää vaatimukset systemaattisten katselmointien määrittelyyn ohjelmiston hankinta-, toimitus-, kehitys-, käyttö- ja yllä-/kunnossapitoprosesseissa. Se kuvaa, miten erityyppiset katselmoinnit on toteutettava. Standardia käytetään yhdessä muiden ohjelmistosuunnittelustandardien kanssa, jotka määrittelevät, mitä katselmoidaan ja milloin. Katselmointi on myös osa vaatimushallintaprosessia.

IEEE Std 830-1998 [8-25] sisältää suositellut lähestymistavat ohjelmiston vaatimusten määrittelyyn. Standardin kohderyhminä ovat ohjelmistojen ostajat ja toimittajat sekä vaatimusspesifikaatioiden laatijat. Standardi esittää hyvän vaatimusspesifikaation kriteereitä, vaatimusmäärittelydokumentin rakenne- ja sisältövaatimukset sekä sisältötemplaten ja liitynnät IEEE 12207:n kuvaamaan prosessiin (Software Life Cycle Processes). Standardi sopii ohjelmiston järjestelmävaatimusdokumentaation laatimiseen, mikä on viimeinen vaihe vaatimusmäärittelyprosessissa. Standardi on erittäin selkeä ja hyvä ohje dokumentaation tuottamiseen.

ISO-standardit

ISO 15504 (SPICE) [8-26] on malli ohjelmistotoimittajan ohjelmistokehitysprosessin arviointiin. Se on samantapainen kuin em. CMMI- malli1. Standardi on kaikkiaan 9-osainen kattaen prosessimallin kuvauksen sekä ohjeet prosessien arviointiin ja parantamiseen. Malli kattaa kaikki vaatimushallintaprosessin osat ml. sidosryhmätarpei-

den tunnistamisen, mutta ei vaadi sidosryhmätarpeiden muuntamista sidosryhmävaatimuksiksi.

NRC-standardit/ohjeet

NRC Regulatory Guide 1.172 [8-27] on ydinvoimalaitosten turvallisuusjärjestelmien atk-ohjelmiston vaatimusspesifikaatioiden laadintaan tarkoitettu standardi Se sisältää kommentteja, täydennyksiä ja soveltamisohjeita joihinkin IEEE 830-1993:n kohtiin. Se painottaa erityisesti vaatimushallinnan osalta jäljitettävyyden hallintaa sekä vaatimusspesifikaation muutoshallintaa. Vaikka IEEE 830 ei ole varsinaisesti tarkoitettu turvallisuusjärjestelmien ohjelmiston laadintaan, se voi opastaa myös siinä. Tämä ohje sellaisenaan ei ole vaatimushallintaan kattava, mutta yhdessä viitestandardien kanssa melko hyvä.

NRC Regulatory Guide 1.97 [8-28] sisältää onnettomuustilanteen jälkeiseen valvontaan tarkoitetuille järjestelmille asetetut vaatimukset. Ohje ei sisällä vaatimushallintanäkökulmaa.

8.2 Havaintoja standardeista vaatimushallintanäkökulmasta

8.2.1 Yleisiä piirteitä

Useimmissa analysoiduissa standardeissa ei vaatimushallintaprosessia kuvata prosessinäkökulmasta, mutta prosessin osia on standardeihin ikään kuin ”sisäänrakennettuina”: Standardeissa on esitetty vaatimuksia sille, mitä on tehtävä tai mitkä ovat osavaiheiden lopputulokset. Ainoastaan CMMI- ja ISO 15504 -standardeissa on vaatimushallinta selkeämmin prosessina esillä.

Systemaattinen vaatimushallintaprosessinäkökulma tuo järjestelmien kehittämiseen ”punaisen langan”, joka ohjaa kehittämistyötä ja varmistaa osaltaan korkealaatuisen lopputuloksen saavuttamista. Vaatimushallinnalla varmistetaan, että lopputulos vastaa sidosryhmien tarpeita.

Kun standardeissa puhutaan vaatimuksista, niin lähes kaikissa yhteyksissä niitä ei erotella sidosryhmä- ja järjestelmävaatimuksiin, ja useimmissa tapauksissa vaatimuksilla tarkoitetaan järjestelmävaatimuksia.

Sidosryhmävaatimusten määrittäminen puuttuu lähes kaikista standardeista. Sidosryhmävaatimusten määrittely on monimutkaisissa järjestelmissä tarpeen aina. Niillä on vaikutusta järjestelmän toiminnan onnistumiseen. Toisaalta yleisissä suunnitteluperiaatteissa on usein otettu huomioon esim. hyviä käyttöliittymäkäytäntöjä, jotka sisältävät käyttäjätarpeita, mutta eivät välttämättä aina kattavasti.

Yksittäisissä vaatimushallinnan elementeissä tunnistettiin seuraavia seikkoja:

- Vaatimustietojen ja niiden elinkaaren aikaisten muutosten hallintaa ei yleensä ole vaadittu.
- Jäljitettävyys on joskus mainittu, mutta sen merkitystä ei erityisemmin ilmaistu.
- Vaatimuslähteistä ja muista vaatimuksiin liittyvistä lisätiedoista (attribuutit) ei ole juurikaan mainintoja.
- Joissakin standardeissa on esitetty suunnitteluvaatimuksia ja -ratkaisuja: miten jokin järjestelmä tai laite on suunniteltava tai toteutettava samoin kuin mitä osia/elementtejä niissä on oltava: nämä voitaisiin mahdollisesti korvata toiminnallisilla & suorituskykyvaatimuksilla, jolloin ne eivät rajoittaisi suunnitteluratkaisuja samassa määrin kuin nyt.
- Vaatimukset ovat usein hyvin yleisluontoisia (tavoitetyyppisiä), jolloin ne eivät aina ole yksikäsitteisesti todennettavissa (esim. ”good documentation shall be provided”, ”automatic testing aids should be available”): yleisluontoisia vaatimuksia voidaan myös purkaa alemmalle tasolle ilman että se rajaa ratkaisuvaihtoehtoja. Tässä suhteessa tavoitepohjainen vaatimusmäärittelylähestymistapa voi olla hyvä keino.
- Standardit soveltuvat hyvin tarkistus- ja muistilistaksi: mitä asioita on muistettava/oltava dokumenteissa.
- Standardien ohjeosioissa on jonkin verran opastusta vaatimusten määrittelyä varten.
- Standardeissa ei edellytetä käytettäväksi systemaattisia vaatimushallintamenettelyitä.
- Standardeissa ei esitetä, millä tavalla johdetaan alemman tason (järjestelmän tai ohjelmiston tms.) vaatimukset ylemmän tason vaatimuksista eikä esitetä laatu- tms. vaatimuksia ylemmän tason vaatimuksille: käytännöissä tulisi varmistaa, että ylemmän tason vaatimusmäärittely on tehty ja että sen laatu täyttää vaatimukset (mikäli jotain puuttuu, se on täydennettävä ennen alemman tason järjestelmän suunnittelun aloitusta).

- Takaisinkytkentää ja iterointia suunnittelussa ei ole esitetty: suunnitteluprosessille leimaa-antavana on peräkkäisajattelu (vesiputousmalli), spiraalimalleja ei esitetty.
- Vaatimushallinnan edellytysten huomiointi puuttuu (esim. käyttö- ja ylläpitokonseptit, sidosryhmäanalyysi, konseptien kehittäminen ja vertailu).
- Hyviä (tarkistuslistanomaisia) suunnittelukäytäntöjä on sisällytetty standardeihin.
- Vaatimusten priorisointi ei ole esillä juuri missään standardissa.

8.2.2 Termeistä ja käsitteistä

Verification- ja validation-termien käyttö on epäyhtenäistä standardeissa:

- Todentaminen (verification, SFS-ISO 9000/2000) on vaatimushallinnassa yläkäsite ja myös osaprosessi, jota tehdään koko järjestelmän toteutusprojektin ajan. Katselmointi, auditointi, analyysi, testaus, demonstrointi ja tarkastus ovat yleisimmin käytettyjä todentamistapoja. Vaatimushallinnassa on myös erotettava
  1) vaatimusten ja muun suunnitteluaineiston todentaminen edeltävän tason vaatimuksia tai tulosaineistoa vastaan sekä
  2) järjestelmän tai sen osien todentaminen vastaavan tason vaatimuksia vastaan.

- Kelpuutus (validation, SFS-ISO 9000/2000) on yläkäsite ja myös osaprosessi. Standardeissa puhutaan usein testauksista erottelematta todentamis- ja kelpuutustestauksia. Varsinaisen kelpuutuksen suorittaa järjestelmän hankkija (esim. laatuorganisaatio) sen ollessa oikeassa käytössä ja käyttöympäristössä. On myös erotettava
  1) vaatimusten ja muun suunnitteluaineiston kelpuutus ylempien tasojen vaatimuksia tai tulosaineistoa vastaan sekä
  2) tuotteen tai sen osien kelpuutus vastaavia vaatimuksia vastaan.

- Kelpoistus-termiä käytetään suomenkielisissä teksteissä samantapaisessa merkityksessä kuin kelpuutus-termiä. Selkeää määritelmää sille ei ole kuitenkaan ole tiedossa.
- Qualification-termin suomalaisena vastineena käytetään usein kelpoistus-termiä, mikä kattaa sekä todentamisen että kelpuutuksen (V&V).

Osa 8 – Standardiviitteet

8-1 CMMI, Nov 2000, CMMI for Systems Engineering/Software/ Engineering/ Integrated Product and Process Development, Version 1.02, CMMI-SE/SW/IPPD.

8-2 EUR Requirements, Vol. 2, Chapter 10, European Utility Requirements for Nuclear Power Plants, Volume 2, Generic Nuclear Island Requirements, Chapter 10, Instrumentation & Control and Man–Machine Interface (Part 1 & Part 2).

8-3 IAEA NS-R-1, IAEA Safety Standards Series No. NS-R-1, Safety of Nuclear Power Plants: Design, Requirements.

8-4 IAEA NS-G-1.1, IAEA Safety Standards Series No. NS-G-1.1, Software for Computer Based Systems Important to Safety in Nuclear Power Plants, Safety Guide, September 2000.

8-5 IAEA NS-G-1.3, IAEA Safety Standards Series, NS-G-1.3, Instrumentation and control systems important to safety in nuclear power plants, Safety Guide, March 2002.

8-6 IEC 60880, Software for computers in the safety systems of nuclear power stations, First edition 1986.

8-7 IEC 60880-2, Software for computers important to safety for nuclear power plants – Part 2: Software aspects of defence against common cause failure, use of software tools and of pre-developed software", First edition 2000-12.

8-8 CEI/IEC (60)964, Design for Control Rooms of Nuclear Power Plants.

8-9 IEC 60780, Nuclear Power Plants – Electrical equipment of the safety systems – Qualification, Second edition 1998-10.

8-10 IEC 60987, Programmed digital computers important to safety for nuclear power stations, First edition 1989-11.

8-11 IEC 61226 Ed. 1.0 b:1993, Nuclear power plants – Instrumentation and control systems important for safety – Classification.

8-12 IEC 61508, Functional safety of electrical/electronic/programmable electronic safety related systems.

8-13 IEC 61513, Nuclear power plants – Instrumentation and control for systems important to safety – General requirements for systems, First edition 2001-03.

8-14 IEC 61839, Nuclear Power Plants – Design of Control Rooms – Functional Analysis and Assignment.

8-15 IEC 62138 ”Nuclear Power Plants – Instrumentation and Control for systems important for safety – Software for computer-based I&C supporting category B or C functions.

8-16 IEC 9126, Information technology – Software product evaluation – Quality characteristics and guidelines for their use.

8-17 ISO/IEC 12207, Software life cycle processes.

8-18 ISO/IEC 15288 System Life-Cycle Processes.

8-19 ANSI/IEEE 1008, IEEE Standard for Software Unit Testing.

8-20 ANSI/IEEE Std 829-1983, IEEE standard for software test documentation.

8-21 IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation.

8-22 IEEE 1074 Draft Standard for Developing Software Life Cycle Processes.

8-23 IEEE 7.4.3.2, March 11, 2003, Draft Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations.

8-24 IEEE Std 1028-1997, IEEE Standard for Software Reviews.

8-25 IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

8-26 ISO 15504 (SPICE project), Software Process Assessment.

8-27 NRC Regulatory Guide 1.172, 1997, Software requirements specifications for digital computer software used in safety systems of nuclear power plants.

8-28 NRC Regulatory Guide 1.97, U. S. Nuclear Regulatory Commission, Regulatory Guide 1.97, revision 3, May 1983. Instrumentation for light-water-cooled nuclear power plants to assess plant and environs conditions during and following an accident.

Liite 8-1 **Standardien arviointiasteikko**

Kuvaa, miten hyvin standardi tukee vaatimushallintaprosessia tai sen osia.
1 = tyydyttävä, 2 = hyvä, 3 = erittäin hyvä, N/A = ei arvioitu.

Elementti / **Standardi**	VH-prosessi kokonaisuutena	Sidosryhmävaatimusten määrittäminen	Järjestelmävaatimusten määrittäminen	Todentaminen	Kelpuutus	Vaatimusten jäljitettävyyden hallinta	Vaatimusten muutosten hallinta	Vaatimusdokumentti	Katselmointi
CMMI	3-								
EUR Requirements, Vol. 2, Chapter 10	1								
IAEA NS-G-1.1	2								
IAEA NS-G-1.3	2-								
IAEA NS-R-1	N/A								
IEC 60880-2	N/A								
CEI/IEC (60)964	N/A								
IEC 60780					2+				
IEC 60880	2								
IEC 60987	1								
IEC 61226 Ed. 1.0	N/A								
IEC 61508	2								
IEC 61513	2-								
IEC 61839	N/A								
IEC 62138	2-								
IEC 9126			1,5						
ISO/IEC 12207	2								
ISO/IEC 15288	2,5								
ANSI/IEEE 1008				2					
ANSI/IEEE Std 829-1983				2					
IEEE 1012				3-	3-				
IEEE 1074 draft	2-								
IEEE 7.4.3.2				2,5	2,5				
IEEE Std 1012-1998				3-	3-				
IEEE Std 1028-1997									2-
IEEE Std 1233									
IEEE Std 830-1998								2,5	
ISO 15504 (SPICE)	3-								

9 Vaatimusmäärittelyn lähestymistapoja

9.1 Tavoitepohjainen lähestymistapa

9.1.1 Yleistä

Turvallisuuteen liittyviltä tietokonejärjestelmiltä edellytetään korkeaa luotettavuutta, ja tämä on voitava myös osoittaa [9-1]. Tarkka ja oikea määrittely siitä, mitä ohjelmiston on tehtävä, on olennainen vaihe järjestelmän kehittämisessä. Silti tänä päivänä on erittäin suuria vaikeuksia ohjelmiston vaatimusten määrittelyssä oikein: on laajasti tunnustettu, että pahimmat ohjelmistovirheet voidaan jäljittää virheellisiin vaatimusmäärittelyihin.

Muodollisia määrittelytekniikoita on ehdotettu tämän ongelman ratkaisuksi ja sen soveltaminen on onnistunut yhä useammin. Siitä huolimatta nykyiset ohjelmistomäärittelytekniikat kärsivät heikkouksista erityisesti kriittisessä vaatimusmäärittelyvaiheessa:

- Rajoittunut kattavuus: Useimmat tekniikat keskittyvät pelkän ohjelmiston määrittelyyn kun tärkeää olisi koko järjestelmän – josta ohjelmisto muodostaa vain osan – määrittely. Ei-toiminnalliset vaatimukset on myös jätetty muodollisten vaatimusmäärittelyiden ulkopuolelle.
- Perusteluiden puute: vaatimusmäärittelyiden ymmärtäminen ilman perusteita on usein vaikeaa.
- Heikko ohjaus: muodollisissa määrittelymenetelmissä pääpaino on määrittelyiden jälkianalyysissä ja hyviä menetelmiä monimutkaisten järjestelmien oikeiden vaatimusmäärittelyiden varsinaiseen *rakentamiseen* ei juuri ole ollut.
- Vaihtoehtotarkastelujen tuen puuttuminen: useimmat määrittelytekniikat eivät mahdollista vaihtoehtoisten määrittelyiden kehittelyä, esittämistä ja vertailuja.

Viitteessä [9-1] esitetty tavoitepohjainen vaatimusmäärittelytapa poistaa osaltaan edellä esitettyjä puutteita.

9.1.2 Tavoitepohjaisen vaatimusmäärittelytavan kuvaus

Tavoitteella tarkoitetaan päämäärää, joka suunniteltavan järjestelmän on saavutettava. Tavoitteet voivat viitata järjestelmän joko toiminnallisiin tai ei-toiminnallisiin ominaisuuksiin ja ne voivat olla hyvin eritasoisia alkaen korkean tason ilmaisuista (”turvallinen ydinvoimalaitos”) ja päätyen yksityiskohtaisiin ilmaisuihin (”turvasignaali oltava pois päältä, kun paine on alle asetusarvon”).

Turvallisuusjärjestelmiin liittyy turvallisuuteen liittyviä tavoitteita, jotka on saavutettava. Tavoitteista voidaan johtaa mm. ohjelmiston vaatimukset ja määrittelyt, ja samalla osoittaa ”takaisin päin” tavoitteiden täyttyminen.

Tavoitepohjainen vaatimusmäärittely lähtee liikkeelle järjestelmän kehittämisen taustalla olevien tavoitteiden tunnistamisella ja etenee purkamalla tavoitteet osatavoitteiksi ja edelleen vaatimuksiksi.

Tavoitteet voidaan ilmaista sekä epämuodollisilla (luonnollinen kieli), puolimuodollisilla että muodollisilla kuvaustavoilla. Viitteen [9-1] esimerkissä on käytetty rinnan sekä epämuodollista että muodollista kuvaustapaa. Tavoitteisiin liitetään yleensä attribuutteja (lisätietoja), esim. vaatimuksen tunniste, nimi ja prioriteetti. Tavoitteiden jäljitettävyyttä ja suhteita muihin tavoitteisiin ja mallinnuselementteihin kuvataan liitynnöillä (linkeillä). Tavoitteet voidaan luokitella useilla eri perusteilla, esim. pääjako toiminnallisiin ja ei-toiminnallisiin samaan tapaan kuin vaatimusten luokittelussakin tehdään.

Tavoitteiden kuvauskieli, attribuutit, luokittelu ja liitynnät ovat tavoitteiden mallintamisvälineitä.

Tavoitemäärittelyn vaiheet:

1. Tavoitteiden tunnistaminen ja kehittäminen
   - Tavoitteet tunnistetaan sidosryhmien ilmaisemista tavoitteista sekä järjestelmän alustavista kuvauksista, esim. aikomuksia koskevista ilmaisuista.
   - Kaikkia järjestelmän tavoitteita ei välttämättä ole aina ilmaistu, vaikka niitä on aina olemassa. Siksi tarvitaan tavoitteiden ”paljastamista” ja dokumentointia.
   - Usein järjestelmää kehitetään olemassa olevan järjestelmän korvaajaksi, ja olemassa olevan järjestelmän ongelmat ja puutteet ovat hyvä lähde uuden järjestelmän tavoitteiden määrittämiselle: kääntämällä ongelmat positiivisiksi saadaan tavoitteita uudelle järjestelmälle.
   - Tavoitejoukkoa voidaan kehittää edelleen skenaarioiden avulla, ristiriitojen poistamisella ja tavoitteiden saavuttamisen esteitä tunnistamalla.
   - Tavoitehierarkiaa kehitetään edelleen alaspäin kysymällä ”Miksi” ja ylöspäin kysymällä ”Miten”. Kehittäminen tapahtuu sekä ylhäältä-alas (top-down) että alhaalta-ylös (bottom-up) periaatteilla. Tällä tavalla saadaan tavoitehierarkia täydennettyä kattavammaksi ja laadukkaammaksi.

2. Objektien kehittäminen
   - Tavoitteisiin liitetään olioita (objekteja), attribuutteja ja liityntöjä
   - Tavoiteilmaisujen perusteella voidaan tunnistaa järjestelmän muuttujia (parametreja), jotka voidaan taas liittää objekteihin (järjestelmän elementteihin)
   - Esimerkki: jäähdytysjärjestelmä [objekti] ja sen muuttuja vedenpaine, joka on samalla jäähdytysjärjestelmän attribuutti.

3. Agenttien mallinnus ja liitynnät
   - Tavoitteita puretaan alemman tason osatavoitteiksi niin kauan kunnes ne voidaan kohdentaa (allokoida) jollekin agentille (agentti on järjestelmän jokin toimija: osajärjestelmä/ laite, ohjelmisto tai ihminen), joka vastaa tavoitteen saavuttamisesta. Kohdentaminen jollekin agentille voidaan tehdä kuitenkin vain, jos ko. agentilla on kyvykkyys täyttää ko. tavoite. Kun osatavoitteen kohdentaminen on tehty, tämä osatavoite muodostaa samalla ko. agentin yksi vaatimusilmaisun.
   - Agenttien liitynnät kuvataan mallilla (esim. kaaviona).

4. Tavoitteiden saavuttamisen esteiden tunnistaminen
   - Esteanalyysillä (obstacle analysis) tunnistetaan niitä esteitä, jotka voivat uhata tavoitteiden saavuttamista. Tavoitteiden negaatioiden avulla saadaan selville esteet (esim. ”ei-turvallinen ohjaus”), joiden pohjalta kehitetään uusia tavoitteita esteen poistamiseksi ja sellaisen ”robustin” järjestelmän aikaansaamiseksi, joka selviää em. esteiden aiheuttamista ongelmista. Esteanalyysi se muistuttaa vikapuumallintamista, jolla tunnistetaan vikojen lisäksi myös toimenpiteet vioista toipumiseksi.

5. Tavoitteiden ristiriitojen käsittely
   - tavoitteet voivat olla keskenään ristiriidassa ja nämä ristiriidat on tunnistettava ja ratkaistava tavoitemäärittelyn yhteydessä
   - ristiriitojen ratkaisussa tavoitemäärittelyitä joudutaan muuttamaan esim. täydentämällä lisäehdoilla tms.
   - ristiriitojen käsittelyä tapahtuu myös tavoitteiden tunnistamisen yhteydessä ja myöhemmin prosessin aikana.

6. Tavoitteiden operaationalistaminen
   - Tavoitteiden operaationalistaminen tuottaa järjestelmän operatiiviset vaatimukset tunnistamalla ne agenteille kohdennettujen tavoitteiden perusteella. Tuloksena saadaan järjestelmän (esim. ohjelmiston) operatiivinen malli, joka määrittelee järjestelmän tuottamat palvelut sekä niiden edellytykset (pre- and postconditions) ja niihin liittyvät vaatimukset.

Tavoitepohjainen vaatimusmäärittely sijoittuu ajallisesti pääosin ennen tavanomaista järjestelmävaatimusten määrittelyvaihetta. Varsinainen

järjestelmävaatimusmäärittely siis alkaa siitä kun tavoitepohjainen analyysi loppuu.

Tavoitemallia voidaan käyttää myös todentamisen ja kelpuutuksen apuvälineenä.

9.1.3 Johtopäätöksiä tavoitepohjaisen menetelmän soveltamisesta

Viitteessä [9-1] on esitetty seuraavia johtopäätöksiä tavoitepohjaisen vaatimusmäärittelyn soveltamisesta turvallisuusjärjestelmälle:

- tavoitepohjainen määrittely katsoo asioita laajemmasta järjestelmänäkökulmasta
- operatiiviset vaatimukset voidaan johtaa ylemmän tason tavoitteista
- tavoitteet muodostavat vaatimusten perustelun ja oikeellisuuskriteerin
- esteanalyysi auttaa tuottamaan korkean luotettavuuden järjestelmiä tunnistamalla järjestelmän vioittumistavat ja tutkimalla vaihtoehtoisia tapoja ongelmien ratkaisemiseksi riittävän ajoissa
- tavoitehierarkia muodostaa hyvän tavan strukturoida myös koko vaatimusmäärittelydokumentti
- yleinen kehys mahdollistaa erilaisten kuvaustapojen käytön eri vaiheissa
- tavoitteiden mallinnus ohjaa koko määrittelyprosessia
- menetelmässä on vielä kehittämistarpeita lähinnä parempien tekniikoiden kehittämiseksi työn tueksi.

9.2 Muodolliset vaatimusmäärittelytavat

9.2.1 Yleistä

Turvallisuuskriittisten ohjelmistojen luotettavuuden varmistaminen on keskeisiä tekijöitä ydinvoimalaitosten sovelluksissa. Eräs lähestymistapa on ohjelmistojen matemaattinen todentaminen. Matemaattinen todentaminen edellyttää muodollisten vaatimusmäärittelytapojen käyttöä. Muodollisia tekniikoita voidaan käyttää vaatimusmäärittelyn lisäksi suunnittelussa, koodin tuottamisessa, ohjelmoitavan logiikan suunnittelussa ja dokumentoinnissa.

Muodolliset vaatimusten määrittely- ja kuvausmenetelmät ovat virheiden välttämistekniikoita, jotka voivat poistaa virheitä järjestelmän vaatimusmäärittely-, määrittely- ja suunnitteluvaiheissa [9-2]. Ne ovat matemaattispohjaisia tekniikoita tuettuna usein työkaluilla ja voivat siten tarjota luotettavan ja tehokkaan tavan järjestelmän mallintamisessa, suunnittelussa ja analysoinnissa.

Muodollinen määrittely tehdään kuvauskielellä, jonka sanasto, syntaksi ja semantiikka ovat muodollisesti määriteltyjä ja jolla on matemaattis–looginen perusta. Muodollisella määrittelyllä voidaan tuottaa yksikäsitteisiä, täydellisiä ja ristiriidattomia vaatimuksia.

Sidosryhmävaatimusten määrittelyssä muodollisia tekniikoita voidaan käyttää vaatimusilmaisujen muotoilussa, analyysissä ja kelpuutuksessa. Järjestelmävaatimusten määrittelyssä niitä voidaan käyttää määrittelyjen jalostamisessa, analyysissä ja muodollisessa todistamisessa (proof).

Muodollisia tekniikoita on käytetty menetyksellisesti monissa projekteissa. Kansainvälisen avaruusaseman (NASA) vaatimusmäärittelyistä sekä turvallisuuskriittisen asejärjestelmän (Rockwell) vaatimusmäärittelyistä on menetelmällä löydetty vakavia virheitä.

Päämotivaatio muodollisten menetelmien käytölle on varmistaa vaatimusten laatuvaatimusten (yksikäsitteisyys, täydellisyys, ristiriidattomuus, todennettavuus, modifioitavuus ja jäljitettävyys) täyttyminen. Muodollisia menetelmiä voidaan käyttää em. laatuvaatimusten varmistamiseen käyttämällä järjestelmän käyttäytymisen kuvaamiseen mallinnusta tai matemaattista kuvauskieltä.

Eräs syy siihen, että muodollisilla menetelmillä saadaan laadukkaita vaatimusmäärittelyjä, on se, että ne pakottavat tekijät tekemään määrittelyt huolellisesti ja määrittelemään täsmällisesti vaatimukset.

9.2.2 Esimerkkejä muodollisten menetelmien soveltamisesta

Esimerkkejä muodollisten menetelmien käytöstä löytyy mm. ilmailun, rautatiejärjestelmien, ydinvoimalaitosten ja lääketieteen järjestelmien kehittämisessä.

Ilmailun alueella SIFT-projektissa muodollista määrittelytapaa käytettiin lentokoneen ohjaustietokoneen turvallisuusvaatimusten määrittelyssä jo 1970-luvulla. Järjestelmän luotettavuusvaatimukset olivat korkeat ja tuloksena järjestelmän luotettavuus todettiin erittäin korkeaksi.

Rolls-Royce and Associates on käyttänyt ydinvoimalaitoksen ohjelmistojen suunnittelussa muodollisia menetelmiä (pääasiassa VDM:ää) [9-2] Kokemusten mukaan muodollisten menetelmien käyttö on kaksinkertaistanut vaatimusmäärittelyyn käytettävän ajan, mutta eliminoinut uudelleensuunnittelun kokonaan. Kokonaisvaikutus näillä on ollut se, että kustannussäästöt ovat olleet jopa puolet aikaisemmista kustannuksista. Pääjohtopäätöksenä on todettu, että suurin ongelma on järjestelmä kokonaisuutena, ei pelkästään ohjelmisto. Yhdistelmä nykyaikaisia menetelmiä ml. muodolliset menetelmät, voi auttaa aikaansaamaan luotettavia turvallisuuskriittisiä ohjelmistoja. Niiden käytöllä saavutetaan parannuksia aikataulujen, kustannusten ja laadun suhteen.

Rautatiejärjestelmien kehittämisessä on muodollisia menetelmiä käytetty esim. Pariisin RER A-linjan SACEM-ohjausjärjestelmän kehittämisessä [9-2]. SACEM-ohjelmisto koostuu 21000 rivistä Modula-koodia, ja siitä on 63 % katsottu turvallisuuskriittiseksi, ja siksi sen tuottamisessa on käytetty muodollista määrittelymenetelmää (Abrial B). Oikeaksi todistaminen tehtiin manuaalisesti. Koko järjestelmän kelpuutukseen käytettiin noin 100 henkilötyövuoden työmäärä.

Kriittisiltä lääketieteen järjestelmiltä edellytetään myös korkeaa luotettavuutta. Hewlett-Packard on käyttänyt muodollista määrittelytapaa (HP-SL) parantaakseen sydämenhoitotuotteidensa laatua [9-3]. Lääketieteen laitteiden ohjelmistovirheiden johdosta on dokumentoitu monia kuolemantapauksia, esim. Therac-sädehoitolaitteella. Washingtonin yliopistollisen sairaalan syövänhoitokeskuksessa on syklotroni-järjestelmän kehittämisessä käytetty myös muodollisia menetelmiä.

9.2.3 Muodollisten menetelmät riippumattomassa todentamisessa ja kelpuutuksessa

Suurten turvallisuuskriittisten ohjelmistojen riippumattomaan todentamiseen ja kelpuutukseen soveltuvat myös kevyet muodolliset menetelmät, joilla voidaan löytää merkittävät virheet ilman että edellytetään täyttä oikeellisuuden todistamista [9-5]. Viitteessä on esitetty käytännön tapaus, jossa käytetään muodollisia menetelmiä avaruusaseman (International Space Station project (ISS)) ohjelmiston vikojen tunnistamiseen, eristämiseen ja korjaamiseen. Kokemuksen mukaan tärkeintä työssä oli muodollisen menetelmän käyttö, jonka kautta saadaan monia hyötyjä.

Tapauksessa sovellettiin seuraavaa nelivaiheista prosessia:

- vaatimusten (vain osa vaatimusmäärittelyistä) uudelleenmuotoilu selkeässä, tarkassa ja yksikäsitteisessä muodossa
- sisäisten ristiriitaisuuksien tunnistaminen ja poistaminen
- vaatimusten testaus
- tulosten palauttaminen vaatimusten laatijoille.

Uudelleenmuotoillut vaatimukset liitettiin state-machine-malliin käyttäen SCR-mallia (Software Cost Reduction) [9-6], jonka työkalun avulla suoritettiin myös mallin staattisten ominaisuuksien testaus. Dynaamiset ominaisuudet testattiin kääntämällä SRC-state-machine-malli PROMELA-malliksi ja käyttäen mallin tarkistajaa (SPIN) sen käyttäytymien tutkimiseen.

Työssä löydettiin merkittäviä ei-yksikäsitteisiä vaatimuksia, puuttuvia vaatimuksia sekä puutteita toimintojen järjestyksen ja ajoituksen määrittelyissä. Työhön käytettävä työmäärä oli melko pieni (ei käsitelty kaikkia vaatimuksia) ja löydökset olivat merkittäviä.

9.2.4 Yhteenvetoa muodollisista menetelmistä

Turvallisuuskriittisten järjestelmien monimutkaisuus kasvaa ja yhä enemmän toiminnallisuutta aikaansaadaan ohjelmistosovelluksilla. Muodollisilla menetelmillä – yhdistettynä muihin vaatimusmäärittelymenetelmiin – voidaan päästä järjestelmiltä vaadittuihin korkeisiin luotettavuusarvoihin. Niiden käytössä on huomioitava seuraavia näkökohtia [9-3]:

- muodollinen määrittely on jo melko laajalti käytettyä ja ymmärrettyä
- automaattinen oikeaksi todistaminen ei vielä kovin kehittynyttä
- muodolliset menetelmät voivat olla kalliita verrattuna perinteisiin vianpoistotekniikkoihin ja kustannustehokkuus on otettava huomioon tapauskohtaisesti
- vaikka muodollisten menetelmien käyttö parantaa järjestelmien luotettavuutta, emme voi mitata, kuinka paljon se vaikuttaa luotettavuuteen.

Ongelmina on esitetty [9-8] mm. seuraavia:

- kuvaustekniikoita on hyvin monta ja sopivan tekniikan valinta on vaikeaa
- kuvauksen vaatima suuri työmäärä
- käytön hyötyjä ei selkeästi nähtävissä
- tekniikat nähdään lähinnä kuvauskielenä eikä liityntää vaatimusmäärittelyprosessiin ole kuvattu
- työkalutuki ei ole riittävä
- muodollista määrittelyä ei koeta mielekkääksi ongelman ratkaisuvaiheessa, jota vaatimusmäärittely on
- muodollinen määrittely ei pysty aina mallintamaan todellista maailmaa.

Huolimatta edellä esitetyistä ongelmista, kypsiä muodollisia menetelmiä voidaan käyttää tuottamaan turvallisempia ohjelmistoja. Jo pelkkä muodollisen määrittelytavan käyttö voi auttaa selkiyttämään vaatimuksia ja yksinkertaistamaan suunnittelua.

Turvallisuuskriittisten järjestelmien tapauksessa on kuitenkin tärkeää tunnistaa käytetyn menetelmän rajoitukset. Muodollisetkaan menetelmät eivät voi tehdä paljon esimerkiksi kaoottisessa ohjelmantuotantoympäristössä, mihin auttavat kunnolliset prosessit ja niiden seuraaminen. Jotta muodollisten menetelmien käyttö hyödyntäisi turvallisuuskriittisten ohjelmistojen tuotantoa, tarvitaan lisäksi standardoidut menetelmät (ja standardit), niitä tukevia työkaluja ja koulutusta niiden soveltamiseen ja käyttöön.

Muodollisia vaatimusmäärittelytapoja voidaan käyttää myös todentamiseen ja kelpuutukseen sekä laajasti sovellettuna, jolloin sen avulla todistetaan määrittelyn oikeellisuus, että kevyemmin sovellettuna, jolla saavutetaan vaatimusmäärittelyn parempi laatu.

9.3 Näkökulmapohjainen lähestymistapa

9.3.1 Yleistä

Monipuolinen tarkastelu järjestelmän vaatimusten määrittelyssä tuottaa järjestelmän sidosryhmiä paremmin tyydyttävän tuloksen. Hyvä vaatimusmäärittelijä tarkastelee aina asioita useista näkökulmista. Näkökulmapohjaisia lähestymistapoja, jotka tunnistavat ja erottavat systemaattisesti eri näkökulmat, käytetään vielä aika vähän.

9.3.2 PREview-menetelmä

Viitteessä [9-4] on esitetty joustava lähestymistapa (PREview), jota voidaan soveltaa monenlaisiin järjestelmiin. Se on kehitetty yhteistyössä avaruus- ja rautatiejärjestelmien kehittämisessä mukana olleiden asiantuntijoiden kanssa. Lähestymistapa painottuu vaatimusten tunnistamis- ja määrittämisvaiheeseen.

PREview-menetelmässä on pyritty poistamaan monien muiden näkökulmamenetelmien puutteita, joista keskeisin on liiallinen rajoittuminen vain yhden näkökulman ympärille. PREview:ssä voidaan määritellä useita järjestelmään parhaiten sopivia näkökulmia.

PREview:ssä kukin näkökulma koostuu seuraavista osista:

- näkökulman *nimi*: identifioi yksikäsitteisesti näkökulman
- *näkökulma* (fokus): näkökulmat jaetaan kolmeen päätyyppiin, vuorovaikutus- (interactor), epäsuora (indirect) ja osaamisalatyyppeihin (domain). Vuorovaikutusnäkökulma sisältää niiden ihmisten tai järjestelmien näkökulmat, jotka ovat suoraan vuorovaikutuksessa järjestelmän kanssa (esim. järjestelmän käyttäjä, kunnossapitäjä). Epäsuora näkökulma pitää sisällään niiden ihmisten, organisaatioiden tai järjestelmien näkökulmat, joilla on jokin intressi järjestelmän suhteen (esim. viranomainen, omistaja). Osaamisalanäkökulmat ovat näkökulmia, jotka sisältävät ko. järjestelmään liittyvää ammatti- ja erityisosaamista (esim. materiaaliosaaminen).
- *mielenkiinnon kohde* (concern): ohjaavat tärkeiden vaatimusten määrittelyä
- *lähteet:* näkökulmaan liittyvien vaatimusten lähteet (ihmiset, roolit, muut järjestelmät, dokumentit)
- *vaatimukset:* näkökulmaan liittyvät vaatimukset, jotka ovat peräisin ao. lähteistä ja vaatimusanalyysistä
- *muutoshistoria:* näkökulman tietojen muutostiedot.

Näkökulmat voidaan joustavasti sovittaa organisaation käytäntöihin. Näkökulmia voidaan käyttää myös vaatimusdokumentoinnin organisointitapana, mikä helpottaa vaatimusten analysointia ja varmistaa, että tärkeää tietoa ei puutu. Ne toi-

mivat myös vaatimusten jäljitettävyyden mekanismina. Vaatimusten kuvaustapojen valinta voidaan tehdä täysin vapaasti ja tarkoituksenmukaisella tavalla.

PREview-prosessi on spiraalimainen, ja se koostuu vaatimusten tunnistamis-, analyysi- ja neuvotteluvaiheista. Tunnistamisvaiheessa tuotetut vaatimukset täydennetään analyysivaiheessa, jossa myös tunnistetaan ristiriitaisuudet ja puutteellisuudet. Ne ratkaistaan vaatimusten neuvotteluvaiheessa. Prosessi voidaan tehdä tarvittaessa useampaan kertaan spiraalimallin mukaisesti. Prosessi voidaan helposti integroida organisaation olemassa olevien menetelmien kanssa vaatimusmäärittelyn ensimmäiseksi vaiheeksi.

Viitteessä [9-4] on esitetty PREview:n soveltamisesimerkkinä rautateiden junasuojausjärjestelmän (Train Control System, TCS) vaatimusmäärittely, jossa menetelmää kokeiltiin. Kokemus osoitti, että menetelmä on hyvin käyttökelpoinen ja joustava, eikä edellytä soveltajiltaan syvällistä menetelmäosaamista.

PREview käsittelee sidosryhmävaatimusmäärittelyn ongelmallisia osa-alueita, joita ovat:

- vaatimusten tarkastelu eri näkökulmista, jonka PREview tekee näkyväksi ja opastaa näkökulmien tunnistamisessa ja johtamisessa
- yksityiskohtien katoaminen yleisluontoisessa menetelmässä, mihin PREview tarjoaa vastalääkkeitä vaatimuslähteiden, näkökulmien ja mielenkiinnon kohteiden tunnistamisen kautta
- epäselvyys siitä, milloin vaatimusmäärittely on syytä lopettaa. Spiraalimalli mahdollistaa vähittäisen täydentämisen eikä tähtää alun perinkään täydelliseen vaatimusmäärittelyyn heti alussa
- vaatimukset eivät ole näkyvissä ja ilmeisiä, ne on ”kaivettava” esiin, ja PREview mahdollistaa tämän mm. vaatimuslähteiden sekä osaamisalanäkökulman – jolla ei ole sidosryhmää – käsittelyn kautta.

9.4 Oliopohjainen lähestymistapa

Oliopohjainen lähestymistapa perustuu järjestelmän mallintamiseen olioina (object) [9-7]. Olio on kohde, joka määritellään attribuuteilla ja siihen liittyvillä operaatioilla. Oliot ovat järjestelmän merkittäviä toimijoita, agentteja ja palvelijoita. Olioita voivat olla esim. laitteet, joiden kanssa järjestelmä on vuorovaikutuksessa, organisaatioyksiköitä, tapahtumia, fyysisiä sijaintipaikkoja, ohjelmistoelementtejä tai ihmisrooleja.

Järjestelmä kuvataan olioiden muodostamana kokonaisuutena, mikä määrittelee sen käyttäytymisen. Ensin tunnistetaan sidosryhmävaatimukset, joiden perusteella rakennetaan oliomalli.

Oliopohjaisen menetelmän yhtenä etuna on se, että siirryttäessä järjestelmäkehityksessä vaatimusmäärittelystä suunnitteluun, sama oliomalli toimii edelleen työvälineenä, eikä ole tarvetta tehdä jyrkkiä muutoksia kuvaamistavassa. Erityistä turvallisuusaspektia menetelmälle ei ole esitetty.

Osa 9 – Viitteet

9-1 Emmanuel Letier and Axel van Lamsweerde, High Assurance Requires Goal Orientation.

9-2 Jeremy T Lanman, Using Formal Methods in Requirements Engineering, 18 November 2002.

9-3 J. P. Bowen, V. Stavridou, Formal Methods And Software Safety.

9-4 I. Sommerville, P. Sawyer and S. Viller, Viewpoints for requirements elicitation: a practical approach.

9-5 Steve Easterbrook and John Callahan, Formal Methods for Verification and Validation of partial specifications: A Case Study.

9-6 C. L. Heitmeyer, B. Labaw, and D. Kiskis, ”Consistency Checking of SCR-Style Requirements Specifications,” Second IEEE Symposium on Requirements Engineering, York, UK, March 27–29, 1995.

9-7 Kotonya and Sommerville, Requirements Engineering, Processes and Techniques, John Wiley & Sons, 1998, ISBN 0-471-97208-8.

9-8 João Baptista da Silva Araújo Júnior M.Sc., Metamorphosis: An Integrated Object Oriented Requirements Analysis and Specification Method.

10 Muita vaatimushallinnan käytäntöjä

10.1 NASAn vaatimushallintaprosessi

NASAn vaatimushallintaprosessi [10-1] on yhdistelmä hyvistä vaatimushallintakäytännöistä. Sen tavoitteena on luoda, dokumentoida, katselmoida laadukas vaatimusjoukko, vahvistaa vaatimusten perustila (baseline) sekä hallita järjestelmän vaatimuksia.

10.1.1 NASAn vaatimushallintaprosessin pääpiirteet

Prosessi on jaettu neljään päävaiheeseen:

1. Sisällön määrittely (scope).
2. Vaatimusten määrittely.
3. Vaatimusten kelpuutus.
4. Perustilan vahvistaminen ja vaatimusten hallinta.

Prosessi on luonteeltaan iteratiivinen tai spiraalimainen. Prosessikuvauksen lisäksi on tehty tarkistuslistoja esim. vaatimusten laadun arviointia varten.

1. Järjestelmän sisällön (scope) määrittely

Lähtökohtana osavaiheelle on, että järjestelmän sidosryhmät on tunnistettu ja että yläjärjestelmä on dokumentoitu ja se on saatavilla. Prosessi on jaettu seuraaviin osittain rinnakkaisiin osiin, joiden tehtäviä on erikseen tarkennettu:

- Järjestelmän tarpeiden, tavoitteiden, olettamuksien, reunaehtojen ja vastuiden määrittely.
- Järjestelmän operatiivisen konseptin määrittely.
- Järjestelmän ulkopuolisten liityntöjen määrittely.
- Sisältömäärittelyn laadinta, kelpuutus sekä riskien tunnistaminen.
- Dokumentointi.

2. Vaatimusten määrittely

Lähtökriteereinä vaiheelle on järjestelmän katselmoitu ja hyväksytty sisällön määrittelydokumentti.

Prosessi on jaettu seuraaviin osittain rinnakkaisiin osiin, joiden tehtäviä on erikseen tarkennettu:

- Vaatimusten keräys ja dokumentointi.
- Vaatimusten perusteluiden dokumentointi.
- Vaatimusten oikean hierarkiarakenteen määrittäminen ja vaatimusten allokointi järjestelmän osille.
- Jäljitettävyyden määrittely ja dokumentointi.
- Todentamistavan ja -vaiheen määrittely.
- Vaatimusdokumentin tuottaminen.

3. Vaatimusten kelpuutus

Lähtökriteerinä on, että vaatimusdokumentin osia on kirjoitettu ja valmiina kelpuutukseen. Prosessi on jaettu seuraaviin järjestyksessä tehtäviin osiin, joiden tehtäviä on erikseen tarkennettu:

1. Vaatimusten kieliasun tarkistaminen.
2. Vaatimusten ”hyvyyden” katselmointi.
3. Vaatimusten priorisointi.
4. Vaatimusten muodollinen (formaali) katselmointi.
5. Vaatimusten muutosten riskien arviointi.

4. Perustilan vahvistaminen ja vaatimusten hallinta

Lähtökriteereinä on, että katselmointitilaisuuksien jälkeen on todettu että vaatimusdokumentin perustila voidaan vahvistaa ja että projektissa on määritelty muutoshallintaprosessi.

Prosessi on jaettu seuraaviin, projektin elinaikana koko ajan tapahtuviin rinnakkaisiin osaalueisiin, joiden tehtäviä on erikseen tarkennettu:

1. Vaatimusten perustilan vahvistaminen projektin konfiguraation hallintasuunnitelman mukaisesti.

2. Vaatimusten muutoksenhallinta.
3. Muutospyyntöjen analysointi.
4. Prosessimittareiden seuranta ja mahdollisten parannusehdotuksien kirjaaminen, lisäkoulutuksen järjestäminen jne.

Prosessi on kuvattu selkeästi [10-1]. Kun se ei ole sidottu mihinkään sovellusalueeseen, vaatimustyyppiin tai työkaluun, vaan on yleispätevä, sitä voidaan pitää hyvänä lähtökohtana kehitettäessä omaa vaatimushallintaprosessia.

10.2 RailTrack, West Coast Route Modernisation Programme (WCRM) [10-2]

Projektin tarkoitus on modernisoida Iso-Britannian läntistä rautatietä Lontoosta Glasgow/Edinburghiin. Projektiin, jonka budjetti on 5.8 miljardia puntaa, kuuluu rautatiekiskojen uusintaa, risteyksien uudelleen mallintaminen, sähkölinjojen vahvistaminen ja parantaminen sekä signaloinnin uudistaminen samalla kuin rautatie on koko ajan henkilö- ja tavaraliikenteen käytössä. Projekti koskee varsin vilkasta liikennettä: 2000 junaa päivässä ja 16 milj. matkustajaa vuodessa.

Lopputuloksen laadun varmistamiseksi käytetään paitsi vaatimushallinnan menetelmiä, myös muita järjestelmäsuunnittelun periaatteita ja käytäntöjä.

Projektin tavoitteena on matkustusajan lyhentäminen 1 t 20 min:iin välillä Lontoo–Glasgow nostamalla junien nopeutta jopa 140 mph. WCRM on määrä valmistua vuoden 2005 loppuun mennessä. Projektin ylivoimaisesti tärkein vaatimus on turvallisuus.

Projektissa sovelletaan Systems Engineering (SE) -lähestymistapaa ja siinä erityisesti systemaattista vaatimushallintaa. Vaatimuksia ja niiden jäljitettävyyttä ylläpidetään DOORS-vaatimushallintatyökalussa, johon on rakennettu useita räätälöityjä käyttöliittymiä vaatimusten analysoinnin helpottamiseksi. Tietokannassa ylläpidetään paitsi vaatimuksia ja jäljitettävyyttä, myös muita vaatimuksiin liittyvää tietoa.

Turvallisuuteen liittyviä järjestelmiä on projektissa mm. automaattinen junansuojausjärjestelmä nopeille linjoille sekä junasuojaus- ja varoitusjärjestelmä.

Vaatimusten selventämiseksi ja järjestelmäsuunnittelun apuna käytetään erilaisia mallinnusmenetelmiä, esimerkiksi:

- Staattisia järjestelmämalleja, kuten kontekstidiagrammeja, toiminnallisia malleja ja ”Event Sequence Charts”
- Dynaamisia malleja, kuten suorituskykymalli (matka-aika, kapasiteetti, häiriö), RAM malli (Reliability, Availability, Maintenance) sekä turvallisuuden riskimalli.

Vaatimushallintatyökaluun on luotu erilaisia graafisia näkymiä jotka helpottavat projektin etenemisen seuranta. Näkymissä visualisoidaan väreillä esim. vaatimusten kelpuutustilaa tai vaatimusten täyttymisastetta.

Järjestelmän todentaminen ja kelpuuttaminen tapahtuu perinteisen V-mallin mukaisesti. Tässä tarvittavat tiedot hallitaan myös DOORS-työkalulla. Todentamisen tulokset kirjataan vaatimustietokantaan. Näin pystytään jäljitettävyyden kautta varmistamaan vaatimusten täyttymistä ja seuraamaan projektin etenemistä.

Vaatimusten kelpuuttamisen oleellisena osana on vaatimushierarkian analysointi ja varsinkin alemman tason vaatimusmäärittelyn kattavuusanalyysi. Tämä on ilman työkaluja lähes mahdotonta, mutta myös vaatimushallintatyökalujen vakionäkymien avulla hyvin vaikeata. Analyysin helpottamiseksi RailTrack on kehittänyt myös tälle tehtävälle omia työkalunäkymiä.

Mallinnuksessa käytettiin jo aikaisemmin käytössä olevia työkaluja, jotka ovat välttämättömiä järjestelmän mallintamisessa. Uutta tässä projektissa oli vaatimushallintatyökalun käyttö, jota progressiivisesti integroitiin suunnitteluprosessiin.

Vaatimushallintatyökalun käyttö mahdollisti helpommin järjestelmällisen lähestymistavan, ja jäljitettävyyden kautta saatiin varmuus että kaikki tärkeimmät elementit oli otettu huomioon. Samalla järjestelmän kokonaiskuva saatiin käyttäjille havainnollisesti näkyviin.

Vaatimushallinnan ja vaatimushallintatyökalun ottaminen käyttöön on vienyt aika paljon aikaa ja resursseja. Toisaalta, vaatimushallinnan tuominen projektiin oli välttämätöntä. Ilman vaa-

timushallintaa asiakkaan vaatimukset tuskin olisivat täyttyneet, eikä projektia olisi saatu pysymään aikataulussa.

10.3 "Älykäs" hankintaprosessi

Iso-Britannian puolustusministeriössä (Ministery of Defense, MoD) on järjestelmähankintojen, ja alihankintojen hallintaa varten kehitetty ns. älykästä hankintaprosessia (smart acquisition), jonka tarkoituksena on parantaa puolustuskykyä tehostamalla kalustohankintatoimintoa ajan, kustannusten ja tehokkuuden suhteen [10-3].

Hankintaprojektin ohjaavana elimenä toimii ns. IPT (Integrated Project Team), jossa hankinnan pääsidosryhmät ovat edustettuina. Projekti seuraa alla kuvassa 3 olevaa pääprosessia, jossa ensimmäisessä konseptivaiheessa tuotetaan sekä sidosryhmävaatimuksia että järjestelmän elinkaarihallinnan suunnitelma.

Arviointivaiheessa tuotetaan järjestelmävaatimuksia, jotka myöhemmin toimivat toimittajasopimuksen perustana. Demonstraatiovaiheessa keskitytään riskien minimointiin tarkentamalla ja lisäämällä vaatimuksia ja suorituskykytavoitteita. Yleensä tähän vaiheeseen kuuluu myös toimittajien valinta.

Valmistusvaiheessa tuotetaan järjestelmä ja todennetaan järjestelmä järjestelmävaatimuksia vastaan. Tämän jälkeen se kelpuutetaan ja otetaan käyttöön.

Prosessin perustana on hyvin perinteinen vaatimus- ja riskienhallinta, kuitenkin keskittyen järjestelmän elinkaarihallintaan.

Tätä prosessia käyttäen on kuitenkin saavutettu merkittäviä parannuksia ja säästöjä, Defence Procurement Agencyn mukaan v. 1998–2008 suunnitelluissa hankinnoissa jo nyt säästetty 230M GBP, ja arvion mukaan tulee vielä 2004 loppuun mennessä säästöjä n. 80M GBP. Myös aikataulujen, budjettien sekä laadun suhteen on saatu huomattavia parannuksia aikaan.

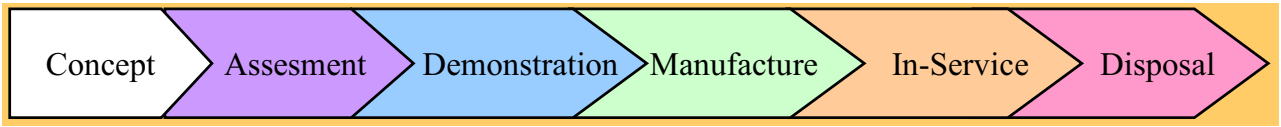
MoD:n hankkeet, joissa on käytetty Acquisition-menetelmää menestyksellisesti v. 2002, olivat mm. Advance Short Range Air–to–Air Missile ja Combat Support Vechicles.

Osa 10 – Viitteet

10-1 Pat Schuler, Tom Schull, Product Requirements Development and Management Procedure, NASA Langley Research Center.

10-2 Brian Halliday, Systems Engineering on the Western Coast Route Modernisation Programme.

10-3 The Acquisition Handbook, Edition 4 – January 2002, http://www.ams.mod.uk/ams/content/handbook/edition4.pdf.



Kuva 3. Järjestelmän elinkaari.

11 Vaihe 2, yhteenveto

Työn toisessa vaiheessa on täydennetty ensimmäisen vaiheen selvitystä ja tietoa järjestelmävaatimusten määrittelytavoista, standardeista sekä lähestymistavoista ja käytännöistä.

Järjestelmävaatimusten määrittelyyn käytettävillä, raportissa esitellyillä menettelyillä saadaan laadittua tai parannettua kattavat ja laadukkaat vaatimusmäärittelyt, joita voidaan käyttää järjestelmän tarjouspyyntödokumenteissa tai suunnittelun ja toteutuksen pohjana. Menettelyistä voidaan valita tarpeen mukaan tilanteeseen sopivat menettelyt. Toiminnallinen ja skenaarioanalyysi sekä Out–of–range-analyysi ovat erityisesti sopivia turvallisuuteen liittyvien vaatimusten tunnistamisessa.

Useimmissa turvallisuuteen liittyvissä standardeissa on vaatimushallintanäkökulma jollain tavalla ”sisäänrakennettuna” tehtävien muodossa, vaikkakaan ei prosessin muodossa. Suurimmassa osassa standardeja vaatimushallintaprosessista puuttuu alkupään sidosryhmävaatimusten määrittelyvaihe sekä vaatimusten jäljitettävyyden ja muutosten hallinta. Kattavimmin vaatimushallintaprosessi on esitetty tarkastelussa varsinaisten ”turvallisuusstandardien” ulkopuolisissa CMMI- ja ISO 15504-standardeissa.

Vaatimushallinnan lähestymistavoista on kuvattu tavoite- ja näkökulmapohjaista sekä muodollisia ja oliopohjaisia määrittelytapoja.

Tavoitepohjainen lähestymistapa sopii hyvin mm. turvallisuuskriittisten järjestelmien vaatimusmäärittelyn ”esivaiheeksi”, jonka jälkeen varsinainen vaatimusmäärittely voidaan tehdä muilla menetelmillä. Sen avulla saadaan kattavat vaatimusmäärittelyt, ja sitä voidaan käyttää myös vaatimusdokumentin strukturointiin. Tavoitemallia voidaan käyttää myös todentamisen ja kelpuutuksen apuvälineenä.

Näkökulmapohjainen lähestymistapa varmistaa kattavien vaatimusmäärittelyiden aikaansaamisen. Raportissa esitetty spiraalimallia noudattava PREview-menetelmä on joustavasti sovitettavissa organisaatioiden käytäntöihin. Menetelmä nostaa esiin piilevät näkökulmat ja varmistaa niihin liittyvien vaatimusten esille saannin. Menetelmää on sovellettu myös turvallisuuskriittisiin järjestelmiin hyvin tuloksin.

Muodolliset kuvaustavat ovat yleistymässä erityisesti turvallisuuskriittisissä sovelluksissa, koska niiden avulla saadaan aikaan korkealaatuisia vaatimusmäärittelyitä. Lisäksi niiden yhteydessä voidaan käyttää automaattista oikeaksi todistamista ohjelmistojen todentamisen ja kelpuutuksen yhteydessä. Ohjelmistojen luotettavuutta voidaan parantaa jo sillä, että osa ohjelmistosta määritellään muodollisella menetelmällä ja todennetaan se. Haittapuolena menetelmien käyttöön on sen vaatima aika ja erityisosaaminen. Työkalujen kehittyessä menetelmien käyttö on lisääntymässä.

Oliopohjainen lähestymistapa pohjautuu olioihin, joiden ominaisuuksien avulla järjestelmä määritellään ja mallinnetaan. Erityistä turvallisuusaspektia menetelmälle ei ole esitetty.

Muista käytännön sovelluksista on esitetty NASAn vaatimushallintaprosessi, Iso-Britannian rautatiemodernisointiprojektissa käytetty vaatimushallinkonsepti sekä Iso-Britannian puolustusministeriön ”älykäs” hankintaprosessi.